



# Universidad de las Ciencias Informáticas

## Facultad 1

### **Solución informática para la gestión del proceso de Ubicación Laboral en la Universidad de las Ciencias Informáticas**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Autor:** Froilán Morales Medina

**Tutores:**

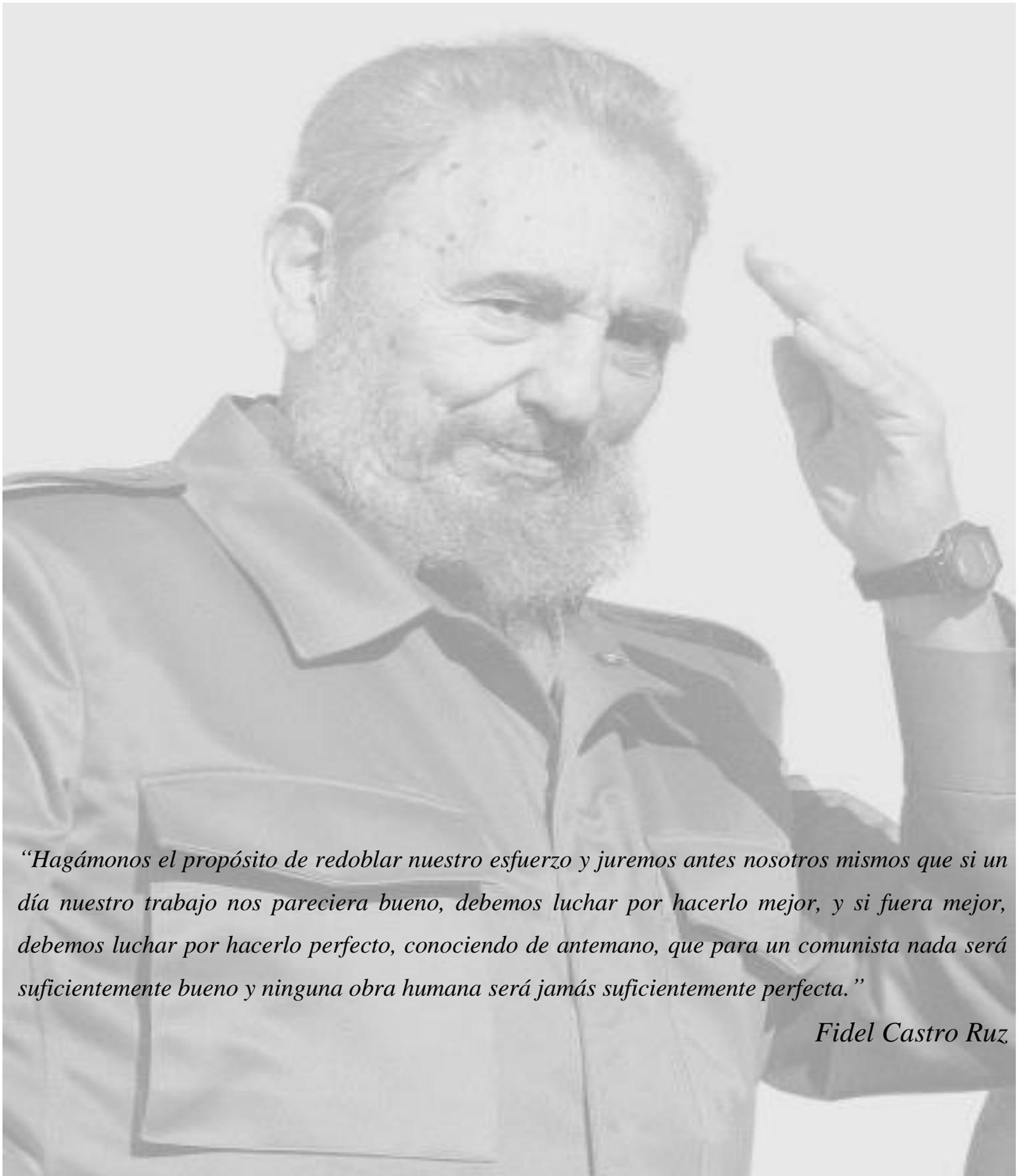
Ing. Hector García Llanes

Ing. Lianet Liben Martínez

Ing. Norges Sánchez Tumbarell

Ing. Odaimys Mercedes Hechavarria Vargas

La Habana, 2014



*“Hagámonos el propósito de redoblar nuestro esfuerzo y juremos antes nosotros mismos que si un día nuestro trabajo nos pareciera bueno, debemos luchar por hacerlo mejor, y si fuera mejor, debemos luchar por hacerlo perfecto, conociendo de antemano, que para un comunista nada será suficientemente bueno y ninguna obra humana será jamás suficientemente perfecta.”*

*Fidel Castro Ruz*

## DECLARACIÓN DE AUTORÍA

Declaro ser autor del presente trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Autorizo a dicho centro para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Froilán Morales Medina

\_\_\_\_\_  
Firma del Autor

Ing. Hector García Llanes

\_\_\_\_\_  
Firma del Autor

Ing. Lianet Liben Martínez

\_\_\_\_\_  
Firma del Autor

Ing. Norges Sánchez Tumbarell

\_\_\_\_\_  
Firma del Tutor

Ing. Odaimys Mercedes Hechavarria Vargas

\_\_\_\_\_  
Firma del Tutor



## **DATOS DE CONTACTO**

### **Hector García Llanes**

Graduado de Ingeniero en Ingeniería Industrial en el año 2005 en el Instituto Superior Politécnico "José Antonio Echeverría". Con experiencia en el diseño de procesos y Dirección estratégica. Se desempeña como Asesor Técnico Docente de la Vicerrectoría de Producción atendiendo el proceso de Ubicación Laboral en la Universidad de las Ciencias Informáticas. Tiene experiencia docente, ha sido Jefe del Departamento Docente de Ciencias Básicas y posee la categoría docente de Asistente. Actualmente se encuentra realizando su doctorado asociado al Centro de Estudios para el Perfeccionamiento de la Educación Superior (CEPES) en la gestión de riesgos en la Planificación estratégica en Instituciones de Educación Superior.

### **Ing. Lianet Liben Martínez**

Graduada de Ingeniera en Ciencias Informáticas en el año 2007 en la Universidad de las Ciencias Informáticas. Con más de seis años de experiencia en análisis de sistemas. Se desempeña como analista de software de la Dirección de Informatización de la Universidad de las Ciencias Informáticas. Con experiencia como docente, posee la categoría docente de Instructor. Actualmente se encuentra cursando la Maestría en Ciencias de la Educación asociado al CEPES.

### **Ing. Norges Sánchez Tumbarell**

Graduado de Ingeniero en Ciencias Informáticas en el año 2007 en la Universidad de las Ciencias Informáticas. Con más de seis años de experiencia en el desarrollo de sistemas y arquitecto de base de datos. Se desempeña como desarrollador de software de la Dirección de Informatización de la Universidad de las Ciencias Informáticas. Con experiencia como docente, posee la categoría docente de Instructor.

### **Ing. Odaimys Mercedes Hechavarría Vargas**

Ingeniera en Ciencias Informáticas. Dos años de experiencia en análisis de sistemas. Analista del Sistema de Gestión Académica de Pregrado perteneciente Dirección de Informatización de la Universidad de las Ciencias Informáticas. Actualmente se encuentra cursando la Maestría en Ciencias de la Educación asociado al CEPES.

## **AGRADECIMIENTOS**

Gracias.

A mis padres y a mi novia por ayudarme día a día en la realización de mis sueños.

A mis amigos por su apoyo incondicional.

A mis tutores por guiarme a lo largo de la elaboración de este trabajo y a todos mis consultantes, en especial a Elizabeth, Jorgito y Emmanuel.

A todos aquellos que han aportado a mi formación en estos cinco años de estudio.

DEDICATORIA

*A Mis Padres*

## **RESUMEN**

El proceso de Ubicación Laboral en la Universidad de las Ciencias Informáticas se realiza cada curso escolar y está dirigido a los estudiantes civiles, graduados del curso regular diurno. Esta tarea es realizada por el Asesor Técnico Docente de la Oficina de Ubicación Laboral perteneciente a la Vicerrectoría de Producción, apoyado por diferentes áreas de la universidad. Actualmente en la realización del proceso existen dificultades en la obtención de la información necesaria para la ubicación laboral, en la impresión de boletas de ubicación laboral y la generación de reportes. A pesar de que se han propuesto sistemas informáticos para la gestión de dicho proceso, estos no solucionan los inconvenientes antes mencionados. Por tal motivo es necesario desarrollar una solución informática con tecnologías libres que integrada al Sistema de Gestión Universitaria facilite la gestión del proceso de Ubicación Laboral en la Universidad de las Ciencias Informáticas, utilizando herramientas, tecnologías y lenguajes libres como: Visual Paradigm, NetBeans, JavaScript, entre otras. Para guiar la solución se utilizó la metodología de desarrollo de software Desarrollo Ágil con Calidad, la que combina las metas y prácticas de las áreas de procesos del nivel 2 de CMMI con las buenas prácticas de dirección y desarrollo ágil de los proyectos de software. La presente investigación arrojó como resultado una solución informática que permite mayor rapidez en la obtención de reportes, almacenamiento seguro de la información y la impresión automática de boletas de ubicación laboral.

## **PALABRAS CLAVE**

Asignación de plazas, solución informática, ubicación laboral.



## ÍNDICE

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	5
1.1 Conceptos asociados.....	5
1.2 Proceso de Ubicación Laboral en Cuba .....	5
1.3 Proceso de Ubicación en la UCI .....	6
1.4 Análisis de sistemas de Ubicación Laboral .....	6
1.4.1 Sistema internacional para la gestión de la ubicación laboral .....	7
1.4.2 Sistemas nacionales para la gestión de la ubicación laboral .....	7
1.4.3 Análisis de los sistemas estudiados .....	10
1.5 Descripción de las tecnologías a utilizar en el desarrollo del Módulo de Ubicación Laboral del subsistema de Egreso.....	11
1.5.1 Marco de trabajo: GUUD 1.0 .....	11
1.5.2 Sistema gestor de base de datos: PostgreSQL 9.1 .....	13
1.5.3 Servidor Web: Apache 2.2.2.....	13
1.5.4 Entorno de desarrollo integrado: NetBeans 7.2 .....	13
1.5.5 Administrador de base de datos: PgAdmin 1.10.12 .....	14
1.5.6 Herramienta de modelado: Visual Paradigm 8.0.....	14
1.5.7 Herramienta para crear prototipos de interfaz: Evolus Pencil 1.3.4.....	14
1.5.8 Herramienta para realizar pruebas del sistema: Apache JMeter 2.3.1 .....	14
1.6 Lenguajes de programación.....	15
1.6.1 Pre-procesador de Hipertexto 5.3.22.....	15
1.6.2 JavaScript 1.2.....	15
1.6.3 Hoja de estilo en cascada 3.....	15
1.6.4 Lenguaje de Marcado de Hipertexto 4.....	16
1.6.5 Lenguaje Unificado de Modelado 2.0 .....	16
1.6.6 Notación para el Modelado de Procesos de Negocio .....	17
1.7 Proceso de desarrollo de software.....	17
1.7.1 Metodología .....	17
1.8 Conclusiones del capítulo .....	18

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	19
2.1  Modelado del proceso de negocio .....	19
2.1.1  Reglas del negocio .....	20
2.2  Propuesta de solución .....	20
2.2.1  Integración con el Sistema de Gestión Universitaria.....	21
2.2.2  Integración con el Sistema de Impresión de Documentos .....	22
2.3  Requisitos del software.....	22
2.3.1  Técnicas de obtención de requisitos.....	22
2.3.2  Requisitos funcionales.....	23
2.3.3  Requisitos no funcionales.....	27
2.3.4  Descripción de Requisitos .....	28
2.4  Descripción de la arquitectura y el diseño.....	31
2.4.1  Arquitectura cliente-servidor .....	31
2.4.2  Patrón de arquitectura .....	32
2.4.3  Modelo de despliegue .....	33
2.4.4  Patrones de diseño.....	35
2.4.5  Patrones de diseño de base de datos .....	37
2.5  Componentes visuales del sistema.....	39
2.5.1  Vistas principales .....	39
2.6  Mapa de Navegación .....	41
2.7  Modelo de datos .....	42
2.8  Conclusiones del capítulo .....	43
CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN .....	44
3.1  Paradigmas de programación .....	44
3.2  Estándar de codificación.....	45
3.2.1  Identación, llaves de apertura y cierre, y tamaño de las líneas.....	45
3.2.2  Convención de nomenclatura.....	45
3.2.4  Estructuras de control.....	47
3.3  Validación de la propuesta de solución.....	47
3.4  Validación de los requisitos.....	48

3.4.1. Criterios de validación de requisitos del cliente .....	48
3.4.2. Técnicas de validación de los requisitos.....	48
3.5 Estrategias de pruebas .....	49
3.5.1 Prueba de unidad .....	49
3.5.2 Prueba de integración .....	53
3.5.3 Prueba de validación .....	54
3.5.4 Prueba del sistema.....	55
3.6 Conclusiones del capítulo .....	58
CONCLUSIONES GENERALES .....	59
RECOMENDACIONES .....	60
BIBLIOGRAFÍA REFERENCIADA.....	61
BIBLIOGRAFIA CONSULTADA .....	63
GLOSARIO DE TÉRMINOS .....	66
ANEXOS .....	67
Anexo 1: Diagrama de proceso de negocio.....	67
Anexo 2: Reglas del negocio.....	69
Anexo 3: Descripción de requisitos .....	71
Anexo 4: Criterios para validar requerimientos del cliente .....	115
Anexo 5: Casos de prueba de caja blanca .....	116
Anexo 6: Pruebas de integración .....	121
Anexo 7: Pruebas de rendimiento .....	124

## ÍNDICE DE FIGURAS

Figura 1: Arquitectura Cliente-Servidor. ....	32
Figura 2: Funcionamiento del MVC en GUUD.....	33
Figura 3: Diagrama de despliegue. ....	34
Figura 4: Caracterización del modelo entidad-atributo-valor. ....	38
Figura 5: Árbol fuertemente codificado.....	38
Figura 6: Áreas de la vista de presentación. ....	39
Figura 7: Áreas de la vista de escritorio. ....	40
Figura 8: Áreas de la vista de gestión de procesos. ....	40
Figura 9: Mapa de navegación.....	41
Figura 10: Modelo físico de datos. ....	42
Figura 11: Identación, llaves de apertura y cierre.....	45
Figura 12: Variables.....	46
Figura 13: Clases.....	46
Figura 14: Funciones. ....	46
Figura 15: Documentación. ....	47
Figura 16: Estructuras de control. ....	47
Figura 17: Resultados de las pruebas de rendimiento para 100 usuarios. ....	56
Figura 18: Proceso de negocio de ubicación laboral parte 1. ....	67
Figura 19: Proceso de negocio de ubicación laboral parte 2. ....	68
Figura 20: Resultado de las pruebas de rendimiento para 1 usuario.....	124
Figura 21: Resultado de las pruebas de rendimiento para 50 usuarios. ....	125

## ÍNDICE DE TABLAS

Tabla 1: Requisitos funcionales. ....	23
Tabla 2: Requisitos no funcionales. ....	27
Tabla 3: Descripción de requisito. Asignar ubicación laboral a estudiante. ....	28
Tabla 4: Casos de pruebas de caja blanca del método obtenerCantidadPersonas. ....	51
Tabla 5: No conformidades por iteraciones caja blanca. ....	52
Tabla 6: Prueba de integración entre las funcionalidades Mostrar facultad y Obtener facultad. ....	53
Tabla 7: No conformidades por iteraciones caja negra. ....	58
Tabla 8: Descripción de requisito. Mostrar comisión de ubicación laboral. ....	71
Tabla 9: Descripción de requisito. Crear comisión de ubicación laboral. ....	72
Tabla 10: Descripción de requisito. Modificar comisión de ubicación laboral. ....	76
Tabla 11: Descripción de requisito. Ver detalles comisión de ubicación laboral. ....	79
Tabla 12: Descripción de requisito. Asociar persona a comisión. ....	80
Tabla 13: Descripción de requisito. Eliminar persona de comisión. ....	83
Tabla 14: Descripción de requisito. Mostrar entidad. ....	84
Tabla 15: Descripción de requisito. Crear entidad. ....	86
Tabla 16: Descripción de requisito. Modificar entidad. ....	90
Tabla 17: Descripción de requisito. Ver detalles entidad. ....	93
Tabla 18: Descripción de requisito. Asociar registro de demanda. ....	94
Tabla 19: Descripción de requisito. Eliminar registro de demanda. ....	97
Tabla 20: Descripción de requisito. Mostrar facultades. ....	98
Tabla 21: Descripción de requisito. Mostrar índice de integralidad de estudiante. ....	99
Tabla 22: Descripción de requisito. Registrar indicadores de integralidad a estudiante. ....	100
Tabla 23: Descripción de requisito. Modificar indicadores de integralidad a estudiante. ....	104
Tabla 24: Descripción de requisito. Ver detalles indicadores integralidad a estudiantes. ....	108
Tabla 25: Descripción de requisito. Mostrar caracterización de estudiante. ....	109
Tabla 26: Descripción de requisito. Mostrar ubicación laboral de estudiante. ....	110
Tabla 27: Descripción de requisito. Modificar ubicación laboral de estudiante. ....	112
Tabla 28: Descripción de requisito. Mostrar resumen de evaluaciones. ....	114
Tabla 29: Criterios para validar requerimientos del cliente. ....	115

Tabla 30: Caso de prueba de caja blanca del método chequearNombreRepetidoModificar.....	116
Tabla 31: Caso de prueba de caja blanca del método asociarRolPersona.....	118
Tabla 32: Prueba de integración entre las funcionalidades registrar indicadores de integralidad a un estudiante y obtener promedio de carrera.....	121
Tabla 33: Prueba de integración entre las funcionalidades asignar ubicación laboral a estudiante y obtener estudiantes foto buscador.....	121
Tabla 34: Prueba de integración entre las funcionalidades asignar prioridad a entidad y obtener división administrativa.....	122
Tabla 35: Prueba de integración entre las funcionalidades asignar estudiantes a grupo de certificación y obtener elementos dado token.....	122



## INTRODUCCIÓN

La evolución de las Tecnologías de la Información y las Comunicaciones (TIC) ha dado paso al desarrollo vertiginoso de la sociedad, permitiéndole al hombre automatizar gran parte de sus actividades. Su influencia en todas las esferas de la sociedad ha permitido no solo el incremento en los resultados de la ciencia, la producción y los servicios, sino también en la educación, donde lo logrado hasta hoy era catalogado como inalcanzable.

En la actualidad las TIC se utilizan de manera generalizada en los centros educacionales, con mayor énfasis en las universidades donde constituyen un elemento fundamental en las tareas de investigación, en el proceso de enseñanza aprendizaje y como apoyo a la administración y dirección universitaria. En Cuba la máxima dirección del Estado y del Gobierno presta especial atención a la utilización de las TIC en las instituciones de educación superior del país. Al respecto el antiguo Ministro de Educación Superior, Dr.C. Fernando Vecino Alegret en la Conferencia Mundial de Educación Superior expresó: "...es propósito de la educación cubana emplear e introducir en la enseñanza universitaria tecnología de punta, con el objetivo de formar un profesional capaz de enfrentar los retos del siglo XXI..." (Alegret, 1995).

Las universidades cubanas además de formar profesionales altamente calificados desarrollan otros procesos dirigidos a los estudiantes; uno de ellos es la asignación de ubicación laboral. En la Universidad de las Ciencias Informáticas (UCI) el proceso de Ubicación Laboral está dirigido a los estudiantes civiles, graduados del curso regular diurno y es iniciado cuando los estudiantes arriban a su último año docente, siendo dirigido por la Oficina de Ubicación Laboral de la Vicerrectoría de Producción. Entre las actividades que se realizan en dicha oficina están contempladas el almacenamiento, control y distribución de las plazas laborales enviadas por el Ministerio de Trabajo y Seguridad Social (MTSS) a la universidad. Actualmente, la oficina presenta dificultades al gestionar el proceso de Ubicación Laboral; pues se realiza en su mayoría de forma manual, lo que provoca un aumento en la ocurrencia de errores y el tiempo de obtención de la información necesaria para la toma efectiva de decisiones por parte de los directivos de la universidad.

Para realizar la asignación de las plazas laborales a los estudiantes, el Asesor Técnico Docente debe conformar un escalafón que definirá el orden de las asignaciones teniendo en cuenta datos de diversos tipos como: nombre, provincia, municipio, e índice de integralidad; actualmente el Asesor Técnico Docente no cuenta con una base de datos de la cual obtener la información actualizada, por lo que debe obtenerla a partir de documentos Excel, Word y correos electrónicos provenientes de diferentes áreas de la



universidad; trayendo consigo pérdida de información, repeticiones de datos o la no asignación de estudiantes a una plaza laboral.

Realizada ya las asignaciones de las plazas laborales a los estudiantes egresados de la UCI, es responsabilidad del Asesor Técnico Docente de la Oficina de Ubicación Laboral realizar la impresión de boletas de ubicación laboral que serán entregadas como constancia de la ubicación asignada. Actualmente la impresión de boletas se realiza por medio de una herramienta informática, desarrollada con tecnologías propietarias, la cual al no encontrarse integrada al Sistema de Gestión Universitaria (SGU) provoca que la información sea importada a través de documentos Excel, causando ineficiencia, poca veracidad de los datos e información no actualizada. Además no permite realizar determinados ajustes a la hora de imprimir, como la visualización previa de la boleta y el cambio de diversos datos necesarios para la impresión.

Otra de las tareas realizadas por el Asesor Técnico Docente de Ubicación Laboral es la obtención de reportes, tanto de la graduación actual como de cursos anteriores, los que son solicitados por los directivos de la UCI, el MTSS y el Ministerio de Educación Superior (MES). Los reportes son confeccionados de forma manual, aumentando el tiempo dedicado para su realización y la cantidad de errores, debido fundamentalmente a la cantidad de estudiantes a los que se les realiza el proceso, los que sobrepasan los 1000 estudiantes por cada curso escolar.

De la situación expuesta anteriormente se deriva el siguiente **problema a resolver**: ¿Cómo facilitar la gestión del proceso de Ubicación Laboral en la UCI?

En consecuencia se define como **objeto de estudio**: el proceso de Ubicación Laboral en las universidades cubanas, enmarcándose en el **campo de acción**: el proceso de Ubicación Laboral en la UCI.

Para dar solución al problema planteado se propone como **objetivo general**: Desarrollar una solución informática con tecnologías libres, para facilitar la gestión del proceso de Ubicación Laboral en la UCI.

Se proponen además como **objetivos específicos**:

1. Caracterizar los principales elementos de la fundamentación teórica de la investigación.
2. Diseñar una solución informática para la gestión del proceso de Ubicación Laboral en la UCI a partir del proceso de desarrollo de software utilizado.
3. Implementar una solución informática para la gestión del proceso de Ubicación Laboral en la UCI a partir del diseño realizado y empleando las tecnologías definidas para este fin.



4. Validar la propuesta de solución mediante la ejecución de pruebas de *software*.

Para el cumplimiento de los objetivos específicos de la investigación se plantean las siguientes **tareas de la investigación**:

1. Diagnóstico del estado del arte de los diferentes sistemas para la ubicación laboral existentes en Cuba y el mundo.
2. Caracterización de las tecnologías a utilizar en el desarrollo de la propuesta de solución.
3. Identificación y validación de los requisitos que debe cumplir la propuesta de solución.
4. Análisis de la propuesta de solución a partir de los requisitos identificados.
5. Descripción de la arquitectura y el modelado de la base de datos.
6. Implementación de las funcionalidades definidas para la propuesta de solución.
7. Integración con el Sistema de Impresión de Documentos del Sistema de Gestión.
8. Caracterización de las pruebas a realizar a la solución.
9. Validación de la propuesta de solución.

Para la realización y culminación de la presente investigación, se hace necesaria la utilización de los siguientes métodos científicos.

## **Métodos teóricos:**

- **Analítico-Sintético:** permite extraer los elementos más coherentes e importantes relacionados con el objeto de estudio, proporcionando un mejor análisis y comprensión de la teoría y documentación relacionada con el tema de investigación.
- **Histórico-Lógico:** permitió hacer un análisis histórico de sistemas similares empleados por otras instituciones cubanas y extranjeras, la metodología, los lenguajes y las herramientas utilizadas en la implementación de la solución.

## **Métodos empíricos:**

- **Entrevistas:** mediante las entrevistas realizadas a los analistas y clientes, se obtuvo información necesaria para comprender el proceso de Ubicación Laboral en su totalidad. Se esclarecieron elementos del diseño, aportando elementos muy importantes para la implementación de la aplicación deseada.
- **Revisión documental:** se pone de manifiesto al revisar la bibliografía existente y consultar información en internet para dar solución a las tareas de investigación.



Con el término del trabajo se logrará que la solución informática para la Ubicación Laboral en la UCI quede completamente documentada y funcional.

## **Aportes de la solución:**

- Mayor rapidez en la obtención de reportes.
- Almacenamiento seguro de la información.
- Reducción del volumen físico de la información.
- Disponibilidad de la información.
- Reducción de tiempo y esfuerzo en el proceso de impresión de las boletas de ubicación laboral.

## **Estructura del documento**

El contenido del presente trabajo de diploma está estructurado en tres capítulos que abarcan todo el proceso para el desarrollo de la solución informática.

**CAPÍTULO 1. Fundamentación teórica:** se expone el estado del arte del objeto de estudio de la presente investigación y se definen conceptos fundamentales asociados al dominio del problema. Se analizan soluciones informáticas existentes en el ámbito nacional e internacional, destinadas a la Ubicación Laboral de estudiantes y se caracterizan las tecnologías a utilizar para el desarrollo de la propuesta de solución.

**CAPÍTULO 2. Descripción de la propuesta de solución:** se realiza el análisis del proceso de Ubicación Laboral de la UCI, quedando definido el flujo actual de dicho proceso y las reglas del negocio. También se describe la propuesta de solución y se especifican los requisitos funcionales y no funcionales con sus descripciones correspondientes. Se definen los patrones de diseño y el patrón de arquitectura a utilizar, además del modelo de datos para el registro de la información a gestionar.

**CAPÍTULO 3. Implementación y validación de la solución:** se describe la implementación de la propuesta de solución teniendo en cuenta los paradigmas de programación utilizados y las técnicas de codificación empleadas. Se validan los requisitos de software y realizan pruebas para validar y verificar el correcto funcionamiento del módulo.



## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se exponen los principales conceptos asociados a la problemática a resolver y se estudian diferentes sistemas informáticos para la ubicación laboral, existentes en el ámbito nacional e internacional. Además se caracterizan las tecnologías y la metodología a utilizar en el desarrollo de la propuesta de solución.

### 1.1 Conceptos asociados

Para lograr una mejor comprensión de la investigación, a continuación se abordan un conjunto de conceptos que están estrechamente relacionados con el dominio del problema.

**Gestión:** Proceso mediante el cual se obtiene, despliega o utiliza una variedad de recursos básicos para apoyar los objetivos de la organización (Gestión de la información en las organizaciones. Principios, conceptos y aplicaciones, 2007).

**Gestión de la información:** “La gestión de la información es todo lo que tienen que ver con obtener la información correcta, en forma adecuada, para la persona indicada, al costo correcto, en el momento oportuno, en el lugar indicado, para tomar la acción precisa” (Woodman L, 1985).

**Software de gestión:** Es un programa que sirve como herramienta, la que es desarrollada especialmente para adecuarse a los diferentes requisitos de las empresas. Las aplicaciones en esta área reestructuran los datos existentes para facilitar las operaciones comerciales o gestionar la toma de decisiones. Además de las tareas convencionales de procesamientos de datos, las aplicaciones de *software* de gestión también realizan cálculo interactivo (Pressman, 2001).

### 1.2 Proceso de Ubicación Laboral en Cuba

Desde inicios de la década de los años 90 se viene aplicando un modelo de perfil amplio en la educación superior cubana. La carrera universitaria constituye una etapa en la formación de un profesional y esta debe ser complementada posteriormente con el adiestramiento laboral. Una vez ubicado el recién graduado comienza un proceso de adiestramiento laboral, regulado por Ley, durante el cual obtendrá la preparación necesaria para desempeñar con éxito la plaza asignada.

La Ubicación Laboral en Cuba es un proceso que se le realiza a cada egresado de la educación superior por igual teniendo en cuenta un grupo de medidas para evaluar y mejorar la incorporación de los jóvenes titulados a la vida laboral y social. En este sentido, se pretende conocer las expectativas y necesidades de



los organismos para la contratación de egresados universitarios, en cuanto a las competencias, las habilidades y los requisitos a tener en cuenta para una óptima inserción laboral.

En la actualidad se ha mantenido la política estatal de asegurar una plaza laboral para cada graduado de los cursos diurnos universitarios, en el momento mismo de su graduación. Ello garantiza la inserción laboral inmediata de esos egresados y su ubicación en atención a las prioridades del país, tomando en cuenta sus méritos, condiciones y resultados integrales. En el proceso de Ubicación Laboral de los egresados participan activamente las Instituciones de Enseñanza Superior (IES), las organizaciones estudiantiles y los Organismos de la Administración Central del Estado (Espí Lacomba y otros, 2009).

### **1.3 Proceso de Ubicación en la UCI**

El proceso de Ubicación Laboral en la UCI comienza al iniciar cada curso escolar y está dirigido a los estudiantes civiles graduados del curso regular diurno. Este proceso es llevado a cabo por la Oficina de Ubicación Laboral de la Vicerrectoría de Producción, que es la encargada de almacenar, controlar y distribuir las plazas laborales, así como realizar la impresión de las boletas de ubicación laboral otorgadas por el MTSS. El proceso es inicializado cuando el Asesor Técnico Docente de Ubicación Laboral elabora el cronograma de actividades para este proceso y lo envía a los directivos involucrados. Luego la comisión para la integralidad de cada facultad analiza a cada estudiante de quinto de la facultad correspondiente y le otorga un índice de integralidad, correspondiente a los resultados obtenidos en las diferentes esferas universitarias. Una vez confeccionado el escalafón de integralidad de la universidad, la comisión para la Ubicación Laboral de la universidad analiza a cada estudiante y le asigna una plaza laboral según el índice de integralidad, la provincia y el municipio de estudiante.

### **1.4 Análisis de sistemas de Ubicación Laboral**

El proceso de Ubicación Laboral debido el gran volumen de información que gestiona resulta una tarea compleja a realizar de forma manual, en la actualidad el desarrollo de las TIC ha facilitado una serie de herramientas informáticas que permiten informatizar dicho proceso. A continuación se describen algunos sistemas informáticos para la ubicación laboral existentes en el ámbito nacional e internacional, con el objetivo de analizar en qué medida dichos sistemas brindan solución a la problemática planteada y su vez tomar experiencias en cuanto a su funcionamiento.



### 1.4.1 Sistema internacional para la gestión de la ubicación laboral

#### Programa de Ubicación Laboral

En la Universidad de Externado de Colombia para lograr la vinculación laboral, las empresas presentan sus ofertas laborales mediante un sitio web o remitiéndolas por un correo electrónico habilitado para ello. Para ingresar al programa, los estudiantes y egresados de la facultad deben mantener sus datos actualizados, en especial el correo electrónico, medio por el cual recibirá los enlaces a las ofertas laborales publicadas.

Para los interesados en brindar ofertas de trabajo a estudiantes y egresados de la Universidad, en este sitio se brinda un formulario de Ingreso de Ofertas Laborales, para que la persona pueda ingresar la oferta laboral con mayor facilidad (Universidad Externado de Colombia, 2013).

#### Funciones principales del sistema:

- Gestión de plazas laborales.
- Gestión estudiantes.
- Gestión egresados.
- Generar reportes.

#### Problemas conocidos en el sistema:

- Gestionar las plazas laborales en ocasiones puede requerir de mucho tiempo y ser tediosa dependiendo de la cantidad de ofertas laborales presentadas por las empresas mediante correo electrónico, las cuales deben ser introducidas al sistema manualmente.

### 1.4.2 Sistemas nacionales para la gestión de la ubicación laboral

#### Trabajo de diploma “Sistema de Gestión de la Ubicación Laboral de los egresados de la Universidad de las Ciencias Informáticas”

Trabajo de diploma donde se estudian los procesos del negocio que se llevan a cabo en el ámbito de Ubicación Laboral en la Dirección de Ingreso y Ubicación Laboral en la Universidad de las Ciencias Informáticas. Para el desarrollo de la aplicación propuesta se emplearon las herramientas siguientes: como Sistema de Administración de Contenidos (*Content Management System* en inglés, abreviado CMS) Drupal, sistema gestor de base de datos MySQL, lenguaje de programación PHP, como metodología de desarrollo de *software Rational Unified Process* (RUP). Se hace uso del Lenguaje



Unificado de Modelado (UML) y Visual Paradigm para la creación de los artefactos generados durante el ciclo de vida del *software* (Baños, y otros, 2008).

## **Funciones principales del sistema:**

- Gestionar usuarios.
- Gestionar roles.
- Gestionar plazas laborales.
- Gestionar propuesta para una plaza laboral.
- Gestionar ubicación laboral.
- Generar reportes.

## **Problemas conocidos en el sistema:**

- La funcionalidad Gestionar usuario, resulta un trabajo tedioso para el administrador del sistema, debido a la cantidad de información de los estudiantes a insertar en la aplicación.
- Las funcionalidades Gestionar plazas laborales resulta una tarea tediosa para el administrador del sistema debido a que las plazas laborales se introducen al sistema de forma manual, de una en una.

## **Trabajo de diploma “Desarrollo del Sistema de Gestión de la Dirección de Ubicación Laboral de la Universidad de las Ciencias Informáticas”**

Trabajo de diploma donde se propone un análisis y diseño del proceso de gestión de Ubicación Laboral de la UCI. Para el desarrollo de la aplicación propuesta se emplearon las herramientas siguientes: como sistema gestor de base de datos MySQL, lenguaje de programación PHP, como metodología de desarrollo de *software* Programación Extrema (XP). Se hace uso del Lenguaje Unificado de Modelado (UML) y Visual Paradigm para la creación de los artefactos que se generan durante el ciclo de vida del *software* (González, y otros, 2012).

## **Funciones principales del sistema:**

- Gestionar estudiante.
- Gestionar organismos.
- Gestionar empresas.
- Gestionar plazas laborales.
- Gestionar ubicación laboral.



- Insertar preubicación.
- Generar reportes.

## **Problemas conocidos en el sistema:**

- La funcionalidad gestionar estudiante, resulta tediosa para el Asesor Técnico, debido a la cantidad de información que debe insertar en la aplicación, además tiene que introducir estudiante a estudiante en la base de datos.
- El índice general de cada estudiante es necesario ingresarlo en el sistema como parte de la funcionalidad gestionar estudiante.

## **Trabajo de diploma “Solución informática para la gestión de Ubicación Laboral en la Universidad de las Ciencias Informáticas”**

Trabajo de diploma donde se estudian los procesos del negocio que se llevan a cabo en el ámbito de Ubicación Laboral en la Oficina de Ubicación Laboral en la Universidad de las Ciencias Informáticas. Para el desarrollo de la aplicación propuesta se emplearon las herramientas siguientes: como marco de trabajo GUUD 1.0, sistema gestor de base de datos PostgreSQL, lenguaje de programación PHP, Java Script 1.2, CSS 3 (Cascading Style Sheets) y HTML 4 (HyperText Markup Language) como servidor de páginas web Apache 2.2, como metodología de desarrollo de *software* XP(Programación Extrema) y Scrum. Se hace uso del Lenguaje Unificado de Modelado (UML) y Visual Paradigm para la creación de los artefactos que se generan durante el ciclo de vida del *software* (Álvarez, 2013).

## **Funcionalidades principales del sistema:**

- Gestionar comisiones.
- Gestionar empresas.
- Gestionar organismos.
- Gestión de roles de comisión.
- Gestionar plazas laborales.
- Gestionar ubicación laboral.
- Gestión tipo de comisión.
- Registrar indicadores de integralidad.
- Generar reportes.

## **Problemas conocidos en el sistema:**



- La funcionalidad Gestionar plazas laborales no permite la visualización del resumen de evaluaciones del estudiante a ubicar.
- La funcionalidad Generar reportes solo posibilita la obtención de los datos relacionados con los estudiantes ubicados por provincias y municipios.

### 1.4.3 Análisis de los sistemas estudiados

La selección de la herramienta más conveniente depende de que esta se ajuste a las necesidades de los clientes y cumpla con todos sus requisitos. En el estudio realizado se detectó que algunos de los sistemas informáticos analizados presentan determinadas funcionalidades requeridas por la Oficina de Ubicación Laboral de la UCI, como son la gestión de plazas laborales, ubicación laboral y la generación de reportes. Estos sistemas de igual forma presentan un conjunto de inconvenientes los que se mencionan a continuación.

El *Programa de Ubicación Laboral* de la Universidad de Externado es un sistema privativo, que está orientado a brindar ofertas laborales y no permite la gestión de comisiones, organismos, empresas y ubicación laboral. El *Sistema de Gestión de la Ubicación Laboral de los egresados de la Universidad de las Ciencias Informáticas* fue desarrollado haciendo uso del CMS Drupal, lo que provoca un alto costo funcional y arquitectónico en cuanto a la integración con el subsistema de Egreso del SGU, además este sistema no permite la gestión de comisiones, organismos y empresas. El *Sistema de Gestión de la Dirección de Ubicación Laboral de la Universidad de las Ciencias Informáticas* no permite la gestión de comisiones y la *Solución informática para la gestión de Ubicación Laboral en la Universidad de las Ciencias Informáticas* a pesar que fue desarrollado con las tecnologías establecida por el Grupo de Arquitectura del Departamento de Desarrollo de la DIN nunca fue desplegado, debido a que sus funcionalidades no satisfacen las necesidades del cliente.

De manera general los sistemas anteriormente mencionados no cuentan con un conjunto de funcionalidades requeridas para la ubicación laboral en la UCI como: la gestión de indicadores de integralidad de los estudiantes, empresas priorizadas, prioridades de las empresas según su importancia para los egresados, la impresión de boletas de ubicación laboral y los reportes generados por estos sistemas no satisfacen las necesidades del cliente. Teniendo en cuenta las deficiencias mencionadas, se decide comenzar la implementación de un módulo integrable al subsistema de Egreso del SGU que permita la gestión del proceso de Ubicación Laboral en la UCI a partir de los elementos relevantes



identificados en estudio de estos sistemas y que responda a los requisitos específicos de la Oficina de Ubicación Laboral.

## **1.5 Descripción de las tecnologías a utilizar en el desarrollo del Módulo de Ubicación Laboral del subsistema de Egreso.**

El módulo de Ubicación Laboral perteneciente al subsistema de Egreso se desarrollará utilizando las tecnologías establecidas por el Grupo de Arquitectura del Departamento de Desarrollo de la Dirección de Informatización de la UCI para el desarrollo del SGU al que se encuentra integrado el subsistema de Egreso. A continuación se describen cada una de las tecnologías utilizadas.

### **1.5.1 Marco de trabajo: GUUD 1.0**

Marco de trabajo GUUD (acrónimo creado con las iniciales de los departamentos del antiguo centro CENIA: Gestión Universitaria, Universidad Digital y Gestión Documental) es la unión del marco de trabajo CodeIgniter con jQuery.

#### **CodeIgniter 1.7.3**

CodeIgniter es un marco de trabajo para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. Su objetivo es permitir el desarrollo de proyectos mucho más rápido de lo que podría si estuviera escribiendo código desde cero, proporcionando un rico conjunto de librerías para tareas comúnmente necesarias, así como una interfaz sencilla y la estructura lógica para acceder a estas bibliotecas. CodeIgniter usa el enfoque del Modelo-Vista-Controlador, que permite una gran separación entre la lógica de negocio y la capa de presentación (EllisLab, 2002-2013).

#### **jQuery 1.7.1**

jQuery es una biblioteca gratuita de JavaScript, cuyo objetivo principal es simplificar las tareas de creación de páginas web responsivas, acordes a lo estipulado en la Web 2.0, la cual funciona en todos los navegadores modernos. Otra de las ventajas de jQuery es que se enfoca en simplificar los scripts y en acceder/modificar el contenido de una página web. Finalmente, jQuery es *software* libre y de código



abierto, posee un doble licenciamiento bajo la Licencia MIT<sup>1</sup> y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos (The jQuery Foundation, 2002).

### GUUD 1.0

GUUD posee las características de los marcos de trabajo conocidos usualmente e incorpora nuevas mejoras y adaptaciones. A continuación se muestra una relación de las principales mejoras y modificaciones que incorpora el marco de trabajo GUUD en su infraestructura.

Del lado del cliente:

- Se implementó una serie de *widgets* para utilizarlos de interfaz de algunos de los *widgets* base de jquery-ui como por ejemplo el *date*, el *tab* (ambos son interfaces de los *widgets* del mismo nombre de jquery-ui) y el *popup* (interfaz del *dialog* de jquery-ui). Además de los ya mencionados, se implementaron *widgets* nuevos entre los que se encuentran: *attach*, *menu*, *message*, *tooltip*, *form* (se construyó con la unión de los *plugins* form6 de JQuery el cual se utiliza para el envío de formularios AJAX<sup>2</sup> y el *validate* utilizado para validar formularios), *grid* (utiliza como *plugin* el jqgrid), *multiselect* (para hacer selecciones múltiples), *navbar* (para la creación de barras de navegación), *tree* (para la creación de árboles) y el *graph* (utiliza la librería Highchart, que se emplea en el desarrollo del módulo en la agrupación funcional Reportes).
- Se implementó un plugin a *JQuery* para el manejo de espacios de nombre e internacionalización.
- Se implementaron funciones comunes para todo el sistema (contenidas en los archivos *core.js* y *common.js*) entre las que se destacan: *loadIn*, *getDataJson*, *createSelect*, *isArray*, *isFunction*, *site\_url*.

Del lado del servidor:

- Se le implementó la programación orientada a aspectos.
- Se le agregó manejo de excepciones y mensajes.
- Se le implementó el *IoC* (*Inversion of Control* en español Inversión de Control) para la interacción entre módulos.
- Se le añadió la característica de la modularidad, o sea que una aplicación pueda dividirse en módulos. CodeIgniter no cuenta con esta posibilidad.

---

<sup>1</sup> Instituto Tecnológico de Massachusetts (*Massachusetts Institute of Technology*).

<sup>2</sup> Acrónimo de Asíncrono JavaScript y XML (*Asynchronous JavaScript And XML*).



- Se añadieron los *plugins export\_pi* (permite exportar a los formatos: PDF<sup>3</sup>, CSV<sup>4</sup> y XLS<sup>5</sup>) e *import\_pi*(permite importar desde archivos en formatos CSV o XSL).

Se añadieron y modificaron ayudantes entre los que se encuentran:

- Añadidos: *template* (brinda la posibilidad de usar plantillas), característica que no posee CodeIgniter. Para esto se añadió también la librería *template*, *assets* (utilizado para la integración en las vistas de JavaScript, CSS, imágenes y el *template*), *grid*, *json*.
- Modificados: *form*, *array*, *security* (Vidal,2012).

### 1.5.2 Sistema gestor de base de datos: PostgreSQL 9.1

PostgreSQL es un potente sistema de base de datos, de código abierto objeto relacional. Funciona en los principales sistemas operativos, incluyendo Linux y Windows. Incluye más de 2008 tipos de datos, incluyendo Integer, Numeric, Boolean, Char, Varchar, Date, Interval, y Timestamp. También es compatible con el almacenamiento de grandes objetos binarios, como imágenes, sonidos o vídeo. Cuenta con interfaces de programación nativas para C / C + +, Java, .Net, Perl, Python, Ruby, entre otros, y documentación de apoyo excepcional. (The PostgreSQL Global Development Group, 2010).

### 1.5.3 Servidor Web: Apache 2.2.2

El servidor Apache es un servidor web HTTP<sup>6</sup> de código abierto, para plataformas GNU/Linux, Microsoft Windows, Macintosh y otras. Es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (Félix Alvaro del Castillo, 2000).

### 1.5.4 Entorno de desarrollo integrado: NetBeans 7.2

NetBeans es un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés *Integrated Development Environment*) disponible para los sistemas operativos Windows, Mac, Linux y Solaris. El proyecto

---

<sup>3</sup> Formato de Documento Portátil (Portable Document Format).

<sup>4</sup> Valores separados por comas (*Comma-Separated Values*).

<sup>5</sup> Extensión para los archivos de hoja de cálculo de Microsoft Excel.

<sup>6</sup> Protocolo de transferencia de hipertexto (Hypertext Transfer Protocol).



NetBeans consiste en un IDE de código abierto y una plataforma de aplicaciones que permiten a los desarrolladores crear rápidamente web, aplicaciones de escritorio y aplicaciones móviles utilizando la plataforma Java, así como PHP, JavaScript y Ajax, Groovy y Grails, y C / C + + (NetBeans Community, 2013).

### **1.5.5 Administrador de base de datos: PgAdmin 1.10.12**

PgAdmin es el administrador de base de datos más popular y rico en características de administración de código abierto. La aplicación puede utilizarse en Linux, FreeBSD, Solaris, Mac OS y las plataformas de Windows para administrar PostgreSQL 7.3 ó superiores que se ejecuta en cualquier plataforma, así como las versiones comerciales y derivados de PostgreSQL como *Postgres Plus Advanced Server* y base de datos *Greenplum* (Comunidad PostgreSQL, 2013).

### **1.5.6 Herramienta de modelado: Visual Paradigm 8.0**

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y código inverso (Visual Paradigm Company, 2013).

### **1.5.7 Herramienta para crear prototipos de interfaz: Evolus Pencil 1.3.4**

Evolus Pencil es un proyecto libre y de código abierto para la creación de diagramas y prototipos de las interfaces de usuario de forma rápida y sencilla. Permite crear diagramas o prototipos ya sea web, *software* de escritorio o cualquier otra interfaz que necesite ser diseñada. Es ideal para diseñadores de prototipos y para desarrolladores de *software* que por lo general diseñan interfaces para presentar a sus clientes. (EVOLUTION SOLUTIONS CO, 2013).

### **1.5.8 Herramienta para realizar pruebas del sistema: Apache JMeter 2.3.1**

Apache JMeter, es una herramienta Open Source realizada en Java que se utiliza para realizar pruebas de rendimiento y resistencia, normalmente para aplicaciones web. JMeter permite realizar simulaciones de carga en el servidor, red o aplicación para comprobar su fuerza y para analizar el rendimiento ante diferentes tipos de sobrecarga (Ing. Jon Gómez García de Garayo, 2010).



## 1.6 Lenguajes de programación

Un lenguaje de programación es un lenguaje artificial que intenta parecerse al lenguaje humano, utilizado para definir una secuencia de instrucciones capaces de ser interpretadas y ejecutadas por una computadora. Este lenguaje establece un grupo de reglas sintácticas y semánticas, las cuales rigen la estructura del programa de computación que se escribe.

### 1.6.1 Pre-procesador de Hipertexto 5.3.22

Pre-procesador de Hipertexto (PHP, por sus siglas en inglés Hypertext Pre-processor) es un lenguaje de código abierto muy popular, especialmente adecuado para el desarrollo web y que puede ser embebido dentro de páginas HTML. Gran parte de su sintaxis fue tomada de C, Java y Perl con un par de características específicas propias de PHP. El objetivo del lenguaje es permitir a desarrolladores web escribir páginas generadas dinámicamente con rapidez (The PHP Group, 2001-2013).

### 1.6.2 JavaScript 1.2

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas embebido en el HTML o llamado desde otro fichero. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos.

Su mayor potencial está dado porque permite realizar validaciones de datos o elementos del lado del cliente reduciendo de esta manera la carga del servidor y las transacciones que se efectúan a través de HTTP (Javier Eguiluz, 2013).

### 1.6.3 Hoja de estilo en cascada 3

Hoja de Estilo en Cascada (CSS, por sus siglas en inglés Cascading Style Sheets) es un lenguaje creado para controlar el aspecto o presentación de los documentos electrónicos definidos con Lenguaje de Marcado de Hipertexto (HTML, por sus siglas en inglés HyperText Markup Language) y Lenguaje de Marcado de Hipertexto Extensible (XHTML, por sus siglas en inglés Extensible Hypertext Markup Language). CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Esto brinda numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo. Además, mejora la accesibilidad del documento,



reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. CSS trabaja especificándole el elemento HTML que se quiere modificar y como se quiere mostrar en el navegador (Yuen, y otros, 2004).

#### 1.6.4 Lenguaje de Marcado de Hipertexto 4

HTML es un lenguaje de marcas hipertextuales, un lenguaje diseñado para estructurar textos para generar páginas *web*. Es extensible, se le pueden añadir características, etiquetas y funciones adicionales para el diseño de páginas *web*, generando un producto vistoso, rápido y sencillo. Es utilizado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>).

HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores *web* y otros procesadores de HTML (Álvarez, 2013).

#### 1.6.5 Lenguaje Unificado de Modelado 2.0

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés Unified Modeling Language) es un lenguaje de modelado de sistemas de *software*. UML representa un modelo estándar para visualizar un dibujo técnico de sistema, que incluye elementos como el actor (que especifica el papel que juega un usuario que interactúa con el sujeto), el proceso de negocio (tareas relacionadas lógicamente para lograr un negocio definido), el componente (encapsula el contenido del sistema), la actividad (tarea que toma lugar para cumplir un contrato de operación), los estatutos del lenguaje de programación, los esquemas de la base de datos y los componentes reusables del *software*.

UML se organiza en dos actividades mayores:

- Diseño del sistema.
- Diseño de objetos.

El principal objetivo de UML, diseño de sistema, es representar la arquitectura de *software*.

El diseño de sistemas y objetos en UML se extiende para considerar el diseño de interfaces, administración de datos con el sistema que se va a construir y administración de tareas para los subsistemas que se han especificado (Pressman, 2001).



### 1.6.6 Notación para el Modelado de Procesos de Negocio

La Notación para el Modelado de Procesos de Negocio (BPMN, por sus siglas en inglés Business Process Modeling Notation) proporciona una notación gráfica para especificar los procesos de negocio en un diagrama de procesos de negocio. Su objetivo es apoyar BPM<sup>7</sup> al proporcionar una notación estándar que sea comprensible para los usuarios del negocio. BPMN provee una notación simple para los flujos, independiente del entorno de implementación.

BPMN proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. Define la notación y semántica de un Diagrama de Procesos de Negocio (Object Managed Group, 2013).

## 1.7 Proceso de desarrollo de software

Para guiar el desarrollo de la solución se utiliza la metodología Desarrollo Ágil con Calidad (DAC), la que ha sido pensada para incluir la especificación de las actividades y tareas de cada una de las fases del ciclo de vida de los proyectos, teniendo en cuenta los procesos de Integración de Modelos de Madurez de Capacidades (CMMI, por sus siglas en inglés Capability Maturity Model Integration) nivel 2 para la UCI. A continuación se describe la metodología mencionada anteriormente.

### 1.7.1 Metodología

La metodología DAC es un proceso de desarrollo de *software* que combina las metas y prácticas de las áreas de procesos del nivel 2 de CMMI con las buenas prácticas de la dirección y desarrollo ágil de proyectos de *software*. Es un proceso colaborativo, recursivo-iterativo, incremental y guiado por procesos y requisitos. Su modelo del proceso es una adaptación del modelo en Cascada a los modelos Programación Extrema y Desarrollo Concurrente. Está enfocado a proyectos pequeños o proyectos grandes divididos en sub-proyectos que desarrollan *software* de gestión basado en componentes.

DAC plantea que el problema una vez identificado y definido debe ser descompuesto en problemas más pequeños, y si es necesario, realizar con estos la misma operación. Cada sub-problema será resuelto mediante un componente y el problema resuelto será el *software* o producto final; por lo que las entregas en DAC son a nivel de iteración, en la que habrá obligatoriamente un incremento del producto a partir de

---

<sup>7</sup> Modelado de Procesos del Negocio(Business Process Modeling)



la solución de un componente del mismo. Al finalizar cada iteración se realiza la integración del componente al producto final (Méndez, 2013). Para una mejor comprensión de la metodología DAC, remitirse al artefacto “*Proceso de Desarrollo de Software*” del expediente del proyecto Egreso.

## **1.8 Conclusiones del capítulo**

En este capítulo se explicaron los principales conceptos asociados al tema y brevemente el proceso de Ubicación Laboral contribuyendo a una mejor comprensión del proceso. El análisis de sistemas homólogos utilizados para gestionar la ubicación laboral en el ámbito nacional e internacional demostró, que estos sistemas no satisfacen en su totalidad las necesidades de la Oficina de Ubicación Laboral de la UCI; pero permitieron adquirir elementos relevantes a tener en cuenta para la concepción de la propuesta de solución. El estudio de las herramientas, lenguajes y metodología definidos por el Grupo de Arquitectura de la DIN permitió la familiarización con los elementos del entorno de desarrollo que se utilizarán en la propuesta de solución.



## CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

En este capítulo se describe las actividades del proceso de Ubicación Laboral de la UCI. Se exponen las principales características con las que deberá contar el módulo Ubicación Laboral del subsistema de Egreso y se describe la integración con el SGU y el Sistema de Impresión de Documentos, quedando identificadas las principales funcionalidades del mismo y los artefactos generados en el análisis y diseño de la solución. Se especifican las técnicas utilizadas para la obtención de requisitos del *software* y se describen los mismos. Además se realiza la descripción de la arquitectura, los patrones de diseño, el modelo de datos y los componentes visuales del sistema empleado en el desarrollo de la propuesta de solución.

### 2.1 Modelado del proceso de negocio

El proceso de Ubicación Laboral en la UCI se lleva a cabo a lo largo de cada curso escolar, regido por procedimientos y legislaciones, los que permiten delimitar cada una de sus actividades. Por tanto al encontrarse este proceso bien definido se prosigue a realizar el modelado del proceso de negocio, el cual se presenta en las **Figuras 18 y 19 del Anexo 1: Diagrama de proceso de negocio** y representa el flujo del proceso de Ubicación Laboral.

#### Flujo del proceso de Ubicación Laboral

A continuación se describe el flujo actual del proceso de negocio:

- Enviar el cronograma de trabajo a todos los directivos involucrados con el proceso de Ubicación Laboral de la UCI.
- Confeccionar y enviar el listado de los estudiantes de quinto año.
- Confeccionar y enviar el listado con la cantera de cada facultad de los estudiantes a realizar el Servicio Social en la UCI.
- Seleccionar integrantes de la comisión de ubicación laboral.
- Seleccionar integrantes de las comisiones de integralidad de cada facultad.
- Constituir la comisión de Ubicación Laboral de la UCI y las de Integralidad a nivel de facultad.
- Divulgar los procedimientos y legislaciones vigentes relacionadas con el proceso de Ubicación Laboral.
- Confeccionar y ejecutar el cronograma de los organismos priorizados.



- Elaborar y enviar el Informe de Disponibilidad de graduados.
- Elaborar y enviar el Plan de distribución de graduados.
- Analizar el Plan de Distribución de graduados enviado por el MTSS.
- Elaborar el pre-listado de la demanda de la UCI.
- Ejecutar cronograma de asambleas de integralidad de cada facultad.
- Analizar y asignar indicadores de integralidad a los estudiantes.
- Confeccionar y enviar el escalafón de cada facultad y las caracterizaciones de los estudiantes.
- Confeccionar el Escalafón de Integralidad de la Universidad.
- Analizar y asignar plazas de ubicación laboral.
- Desarrollar las Asambleas de Ubicación Laboral de las Facultades.
- Imprimir y entregar las boletas de ubicación laboral a la Secretaría Docente de cada Facultad.
- Entregar las boletas no procesadas e Informe sobre el uso de las mismas a la DPTSS.
- Confeccionar el Informe sobre el proceso de Ubicación Laboral en la Universidad y el listado final de Ubicación Laboral.

Es muy probable que en la realización de estas actividades existan duplicaciones de archivos y errores, ya que gran parte del trabajo se realiza de forma manual, trayendo consigo la existencia de errores que tienden a afectar directamente el proceso de Ubicación Laboral.

## 2.1.1 Reglas del negocio

Las reglas del negocio son restricciones que las organizaciones tienen definidas y son vitales para lograr sus objetivos. El proceso de Ubicación Laboral presenta diversas reglas que permiten que este proceso se realice con la mayor calidad posible. En el **Anexo 2: Reglas del negocio** se describen las reglas definidas por la Oficina de Ubicación Laboral de la UCI.

## 2.2 Propuesta de solución

El módulo de Ubicación Laboral formará parte del subsistema de Egreso del SGU, este permitirá la gestión del proceso de Ubicación Laboral en la UCI, manteniendo la información de forma segura, accesible y disponible para la toma efectiva de decisiones por parte de los directivos de la universidad. Además contará con una interfaz sencilla e intuitiva, lo cual facilitará la interacción de los usuarios con la



aplicación, sin necesidad que estos tengan avanzados conocimientos informáticos. A continuación se explican las agrupaciones funcionales por la que estará constituida la propuesta de solución.

- ❖ **Diseño de Ubicación Laboral:** en esta agrupación funcional se gestionan las comisiones de ubicación laboral y de integralidad. Además de las entidades, a las cuales se les asignaran los egresados de la universidad.
- ❖ **Ubicación Laboral:** en esta agrupación funcional se gestionará la Ubicación de las plazas laborales y los indicadores de integralidad de los estudiantes.
- ❖ **Configuración:** en esta agrupación funcional se realizará toda la configuración necesaria para llevar a cabo la gestión del proceso de Ubicación Laboral.
- ❖ **Boletas:** para poder realizar exitosamente la impresión de boletas de ubicación laboral es necesario establecer un conjunto de parámetros que son requeridos por el Sistema de Impresión de Documentos del SGU, es en esta agrupación funcional donde se registrarán dichos parámetros.
- ❖ **Reportes:** en esta agrupación funcional se obtendrán los reportes Ubicación Laboral necesarios para la toma efectiva de decisiones por parte de los directivos de la universidad.

## 2.2.1 Integración con el Sistema de Gestión Universitaria.

El módulo de Ubicación Laboral se integrará al SGU lo que permitirá el uso de las facilidades provistas por la integración, así como la reutilización de varios componentes ya implementados por otros sistemas que conforman el SGU. La integración posibilitará la uniformidad de la arquitectura, la centralización de archivos e información y la eliminación de redundancias en el código. Se integrará al Núcleo del SGU con los módulos de Seguridad, Estructura y Composición, Configuración, Documentos Acreditativos, así como al Sistema de Gestión Académica de Pregrado con los módulos de Control Docente, Tesis y Títulos, Carrera y Personal y Secretaría. A continuación se describe la integración con cada uno de los módulos:

- ❖ **Seguridad:** permitirá controlar el acceso al sistema, definiendo para cada usuario el rol que ocupará en la solución y las funcionalidades a las que tendrá acceso dentro del módulo de Ubicación Laboral. El mismo está implementado a nivel central para todo el SGU, garantiza el acceso a la información dado los niveles de privilegio de cada usuario, haciendo uso de la arquitectura sobre la cual está desarrollado el sistema.
- ❖ **Configuración:** permitirá la obtención de las provincias y los municipios, necesarios para filtrar las búsquedas o para registrar algún elemento donde sean necesarios estos criterios.



- ❖ **Estructura y Composición:** permitirá la obtención de las estructuras necesarias para la creación de comisiones y para la consulta de información de los estudiantes en dependencia de la estructura en la que se encuentre ubicado.
- ❖ **Documentos Acreditativos:** permitirá la configuración de los grupos de certificación necesarios para la impresión de boletas de ubicación laboral.
- ❖ **Control Docente:** permitirá consultar la información referente al índice académico de los estudiantes, elemento necesario en el proceso de Ubicación Laboral.
- ❖ **Personal y Secretaría:** permitirá consultar datos de los estudiantes y profesores, como los grupos administrativos, fotos, etc.
- ❖ **Carrera:** permitirá la obtención de la información referente a la carrera que cursan los estudiantes, necesarios para el resumen de evaluaciones y la impresión de boletas.
- ❖ **Tesis y Títulos:** mediante este módulo se obtendrán algunos datos de los estudiantes, como la vía de ingreso a la universidad.

## 2.2.2 Integración con el Sistema de Impresión de Documentos

El módulo de Ubicación Laboral además de integrarse con los módulos anteriormente mencionados, se integrará con el Sistema de Impresión de Documentos del SGU, al cual le proporcionará la información necesaria para realizar la impresión de las boletas de ubicación laboral de los estudiante egresados de la UCI de forma automática, según el diseño establecido para este tipo documento. El Sistema de Impresión de Documentos obtiene la información por medio del módulo Documentos Acreditativos pertenecientes al núcleo del SGU, en el que se registrarán los datos de los grupos de certificación.

## 2.3 Requisitos del software

Un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de este. En el otro extremo, es una definición detallada y formal de una función del sistema (Sommerville, 2004).

### 2.3.1 Técnicas de obtención de requisitos

La obtención de requisitos es el proceso mediante el cual los interesados en un sistema de *software* descubren, revelan, articulan y entienden con diagramas, matrices y métodos de evaluación para extraer los requisitos esperados e intentar obtener requisitos innovadores (Pressman, 2005). A continuación se



describen las técnicas utilizadas durante el proceso de desarrollo del *software* para recopilar los requisitos de las actividades de gestión de ubicación laboral de los egresados de la UCI:

- Entrevista: se realizaron entrevistas no formales al Asesor Técnico Docente de la Oficina de Ubicación Laboral, lo que posibilitó la interacción con el especialista en el área a tratar, para arribar en conjunto a la definición de los requisitos del sistema.
- Sistemas existentes: se realizó un análisis a distintos sistemas ya desarrollados que están relacionados con la Ubicación Laboral. Esto permitió la obtención de elementos necesarios para la modelación y el diseño de prototipos de la propuesta de solución. Algunas de las funcionalidades presentes en estos sistemas sirvieron de base para la propuesta de solución.

## 2.3.2 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, la manera en que este debe reaccionar a entradas de datos y como debe comportarse en situaciones particulares. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville, 2004).

En la siguiente tabla se muestran los requisitos funcionales (RF) identificados para el desarrollo de la solución:

Tabla 1: Requisitos funcionales.

Número	Nombre	Descripción	Prioridad para el cliente	Complejidad
RFUL1	Mostrar comisión de ubicación laboral.	Permite mostrar las comisiones de ubicación laboral.	Alta	Media
RFUL2	Crear comisión de ubicación laboral.	Permite crear una comisión de ubicación laboral.	Alta	Media
RFUL3	Modificar comisión de ubicación laboral.	Permite modificar una comisión de ubicación laboral.	Media	Media
RFUL4	Ver detalles de comisión de ubicación laboral.	Permite ver los detalles de una comisión de ubicación laboral.	Baja	Baja
RFUL5	Asociar persona a comisión.	Permite asociar a una persona como miembro de una comisión.	Alta	Alta

# Descripción de la propuesta de solución



RFUL6	Eliminar persona de comisión.	Permite eliminar una persona de una comisión.	Alta	Media
RFUL7	Mostrar entidad.	Permite mostrar las entidades almacenadas en el sistema.	Alta	Media
RFUL8	Crear entidad.	Permite crear una nueva entidad.	Alta	Media
RFUL9	Modificar entidad.	Permite modificar una entidad.	Alta	Media
RFUL10	Ver detalles de entidad.	Permite ver los detalles de una entidad.	Baja	Baja
RFUL11	Asociar registro de demanda.	Permite asociar un registro de demanda a una entidad.	Alta	Alta
RFUL12	Eliminar registro de demanda.	Permite eliminar un registro de demanda a una entidad.	Media	Media
RFUL13	Mostrar facultades.	Permite mostrar las facultades de la universidad.	Media	Media
RFUL14	Mostrar indicadores de integralidad de estudiante.	Permite mostrar los indicadores de integralidad de los estudiantes.	Alta	Alta
RFUL15	Registrar indicadores de integralidad a estudiante.	Permite introducir cada uno de los valores de los indicadores de integralidad a un estudiante.	Alta	Media
RFUL16	Modificar indicadores de integralidad a estudiante.	Permite modificar cada uno de los valores de los indicadores de integralidad a un estudiante.	Media	Media
RFUL17	Ver detalles de indicadores de integralidad a estudiantes.	Permite ver los detalles de los indicadores de integralidad de un estudiante.	Baja	Baja
RFUL18	Mostrar caracterización de estudiante.	Permite mostrar la caracterización de un estudiante en un documento pdf.	Alta	Baja
RFUL19	Mostrar ubicación laboral de estudiante.	Permite mostrar la ubicación laboral de un estudiante.	Alta	Media
RFUL20	Asignar ubicación laboral a estudiante.	Permite asignar la ubicación laboral de un estudiante.	Media	Alta
RFUL21	Modificar ubicación laboral de	Permite modificar la ubicación laboral	Media	Alta



	estudiante.	de un estudiante.		
RFUL22	Mostrar resumen de evaluaciones.	Permite mostrar el resumen de evaluaciones de un estudiante de toda la carrera.	Alta	Alta
RFUL23	Mostrar rol de comisión.	Permite mostrar el rol de una comisión.	Media	Media
RFUL24	Crear rol de comisión.	Permite crear un nuevo rol para una comisión.	Alta	Media
RFUL25	Modificar rol de comisión.	Permite modificar el rol de una comisión.	Media	Media
RFUL26	Ver detalles de rol de comisión.	Permite ver los detalles del rol de una comisión.	Baja	Baja
RFUL27	Mostrar tipo de comisión.	Permite mostrar los tipos de comisión existentes.	Alta	Media
RFUL28	Crear tipo de comisión.	Permite crear un nuevo tipo de comisión.	Alta	Media
RFUL29	Modificar tipo de comisión.	Permite modificar un tipo de comisión.	Media	Media
RFUL30	Ver detalles de tipo de comisión.	Permite ver los detalles de un tipo de comisión.	Baja	Baja
RFUL31	Mostrar organismo.	Permite mostrar los organismos existentes en el sistema.	Alta	Media
RFUL32	Crear organismo.	Permite crear un nuevo organismo.	Alta	Media
RFUL33	Modificar organismo.	Permite modificar un organismo.	Media	Media
RFUL34	Ver detalles de organismo.	Permite ver los detalles de un organismo.	Baja	Baja
RFUL35	Mostrar entidades priorizadas.	Permite mostrar las entidades priorizadas.	Baja	Media
RFUL36	Crear entidad priorizada.	Permite crear una entidad priorizada.	Alta	Media
RFUL37	Eliminar entidad priorizada.	Permite eliminar una entidad priorizada.	Alta	Media
RFUL38	Asignar prioridad a demanda.	Permite asignarles prioridades a las demandas.	Media	Media



RFUL39	Mostrar grupos de certificación.	Permite mostrar los grupos de certificación registrados en el sistema.	Media	Media
RFUL40	Crear grupo de certificación.	Permite crear un nuevo grupo de certificación.	Alta	Media
RFUL41	Modificar grupo de certificación.	Permite modificar un grupo de certificación seleccionado.	Alta	Media
RFUL42	Ver detalles de grupo de certificación.	Muestra los detalles de un grupo de certificación seleccionado.	Media	Media
RFUL43	Asignar estudiantes a grupo de certificación.	Permite asignar estudiantes a un grupo de certificación.	Media	Media
RFUL44	Registrar datos generales de boletas.	Permite registrar los datos generales de las boletas de graduado referentes a un grupo de certificación.	Alta	Media
RFUL45	Mostrar reporte de estudiantes asignados por organismo.	Permite mostrar un listado con la cantidad de estudiantes asignados por organismo.	Media	Media
RFUL46	Mostrar reporte de estudiantes asignados por provincia.	Permite mostrar un listado con la cantidad de estudiantes asignados por provincias.	Media	Media
RFUL47	Mostrar reporte de plazas ofertadas por organismos y entidades.	Permite mostrar un listado con la cantidad de plazas ofertadas por organismos y entidades.	Media	Media
RFUL48	Mostrar reporte de plazas ofertadas por provincias y municipios.	Permite mostrar un listado con la cantidad de plazas ofertadas por provincias y municipios.	Media	Media
RFUL49	Mostrar reporte de distribución de estudiantes UCI por facultad.	El requisito permite mostrar un listado con la cantidad de estudiantes por facultad que realizarán el servicio social en la UCI.	Media	Media



## 2.3.3 Requisitos no funcionales

Los requisitos no funcionales (RNF) son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Estos a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema (Sommerville, 2004). A continuación se muestran los requisitos no funcionales que debe cumplir el módulo de Ubicación Laboral:

Tabla 2: Requisitos no funcionales.

<b>Usabilidad</b>	
RNF1	Desarrollar una solución web integrada al subsistema de Egreso.
RNF2	La aplicación debe estar dirigida a registrar la información referente al proceso de Ubicación Laboral.
RNF3	El sistema debe presentar una interfaz que permita la fácil interacción por parte de los usuarios, los cuales deben poder acceder de manera rápida y efectiva a la información solicitada.
<b>Seguridad</b>	
RNF4	Solo se mostrarán a los usuarios aquellas acciones o informaciones a las que por su responsabilidad o rol dentro del negocio necesitan.
<b>Disponibilidad</b>	
RNF5	El sistema estará disponible las 24 horas del día y los siete días de la semana.
<b>Eficiencia</b>	
RNF6	El tiempo de respuesta del sistema debe ser como máximo de 5 segundos y soportar una conexión simultánea de hasta 100 usuarios.
<b>Software</b>	
RNF7	El lenguaje de programación a utilizar es PHP 5.3.22.
RNF8	Como IDE se emplea NetBeans 7.2.
RNF9	Como servidor web se utiliza Apache 2.2.2.
RNF10	El diseño de la base de datos se realiza con Visual Paradigm 8.0.
RNF11	El sistema gestor de bases de datos debe ser PostgreSQL 8.4.1.
RNF12	Para el uso del sistema se requiere una PC cliente con el navegador web Mozilla Firefox 15 o superior para el uso de la aplicación web y la máquina virtual de java JDK 6 para el uso del Sistema de Impresión de Documentos.
<b>Hardware</b>	
RNF13	Para el desarrollo de la solución se requiere como mínimo una PC con los siguientes componentes de



	hardware: Procesador Intel Celeron M o superior, de 2 GB RAM (Read Aleatory Memory), 120 GB disco duro.
RNF14	Para el uso del sistema se requiere una PC cliente con los siguientes componentes de <i>hardware</i> : Procesador Pentium 4 o superior, 512 MB de RAM y 1 GB disco duro disponible como mínimo.
RNF15	Para alojar la solución se requiere una PC servidor con los siguientes componentes de <i>hardware</i> : Procesador de 8 núcleos o superior, 4 GB de RAM y 250 GB disco duro disponible como mínimo.
<b>Soporte</b>	
RNF16	El sistema se regirá por el estándar de código documentado en el artefacto “ <i>Estándar de codificación</i> ” del expediente del proyecto Egreso.
RNF17	Se precisa que la documentación del sistema esté actualizada en todos los aspectos, fases de trabajo y ciclos de desarrollo del mismo, permitiendo un respaldo tanto ingenieril como legal del desarrollo de dicho sistema.
<b>Restricciones de diseño</b>	
RNF18	El <i>software</i> deberá ser desarrollado en su totalidad con tecnologías y componentes de código abierto.

## 2.3.4 Descripción de Requisitos

La descripción de requisitos es la base que permite verificar si se alcanzaron o no los objetivos establecidos en el proyecto, ya que estos son un reflejo detallado de las necesidades de los clientes o usuarios del sistema (GIRALDO, 2007). A continuación, se muestran uno de los requisitos descritos. Para consultar las descripciones de requisitos restantes, ver **Anexo 3: Descripción de requisitos**.

Tabla 3: Descripción de requisito. Asignar ubicación laboral a estudiante.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL20	Asignar ubicación laboral a estudiante.	El requisito permite asignar una plaza a un estudiante. El requisito comienza cuando el administrador del sistema, el Asesor Técnico Docente de la Oficina de Ubicación Laboral o la rectora selecciona del módulo Ubicación Laboral del subsistema Egreso, la opción Asignar plaza a estudiante del menú de funcionalidades Ubicación Laboral. Se muestran dos listados, el primero con el listado de	Alta	Alta



		<p>las demandas organizadas por prioridad y el segundo muestra el listado de estudiantes de quinto año.</p> <ol style="list-style-type: none"> <li>1. En el listado de entidades se muestran los datos: prioridad, nombre de la entidad, provincia, municipio, disponibilidad de plazas. Además se muestra un buscador y un filtro de búsqueda por los siguientes criterios: provincia, municipio y organismo.</li> <li>2. En el listado de estudiantes se muestran los datos: nombre y apellidos del estudiante, índice académico, integralidad, índice general y en el área de íconos internos la opción: Resumen de evaluaciones. Además se muestra un buscador y un filtro de búsqueda por los siguientes criterios: provincia, municipio, facultad y grupo.</li> </ol> <p>El usuario selecciona la entidad deseada y el estudiante a asociar a la misma, luego presiona el botón Aceptar, y se muestra el mensaje: “La acción se ha realizado satisfactoriamente”, guardándose para cada estudiante la plaza asignada.</p>		
<p><b>Prototipo</b></p>				



	<div style="text-align: right; margin-bottom: 10px;">☰</div> <h3 style="margin: 0;">Asignar ubicación laboral a estudiante</h3> <div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 45%;"> <input type="text" value="nombre"/> <input type="button" value="Buscar"/> </div> <div style="width: 45%;"> <p style="margin: 0;"><b>Filtrar búsqueda:</b></p> <input type="text" value="-Seleccione-"/> </div> </div> <div style="margin-top: 10px;"> <p style="margin: 0;"><b>Entidades</b> <span style="float: right;">Cantidad por página <input type="text" value="5"/></span></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">Prioridad</th> <th style="width: 20%;">Entidad</th> <th style="width: 20%;">Provincia</th> <th style="width: 20%;">Municipio</th> <th style="width: 15%;">Disponibilidad</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/></td> <td>1 UCI</td> <td>La Habana</td> <td>La Habana</td> <td>40</td> </tr> <tr> <td><input type="radio"/></td> <td>2 MININT</td> <td>La Habana</td> <td>La Habana</td> <td>10</td> </tr> <tr> <td><input type="radio"/></td> <td>3 Emp. Agropecuaria</td> <td>Matanzas</td> <td>Matanzas</td> <td>6</td> </tr> <tr> <td><input type="radio"/></td> <td>4 Emp. Petróleo</td> <td>Holguín</td> <td>Holguín</td> <td>1</td> </tr> </tbody> </table> <p style="text-align: center; margin-top: 5px;"> <input type="button" value="«"/> <input type="button" value="Página 1 de 2"/> <input type="button" value="»"/> <span style="float: right;">Resultados encontrados 8</span> </p> </div> </div> <div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 45%;"> <input type="text" value="nombre, apellidos"/> <input type="button" value="Buscar"/> </div> <div style="width: 45%;"> <p style="margin: 0;"><b>Filtrar búsqueda:</b></p> <input type="text" value="-Seleccione-"/> </div> </div> <div style="margin-top: 10px;"> <p style="margin: 0;"><b>Estudiantes</b> <span style="float: right;">Cantidad por página <input type="text" value="5"/></span></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 30%;">Nombre y apellidos</th> <th style="width: 15%;">Índice académico</th> <th style="width: 10%;">Integralidad</th> <th style="width: 15%;">Índice general</th> <th style="width: 10%;"></th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td> Abel Pérez</td> <td>5.23</td> <td>1.2</td> <td>5.23</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td> Laura Mola</td> <td>5.01</td> <td>1.2</td> <td>5.01</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td> Carlos Martínez</td> <td>4.98</td> <td>1.0</td> <td>4.98</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td> Pedro Díaz</td> <td>4.50</td> <td>0.96</td> <td>4.50</td> <td></td> </tr> </tbody> </table> <p style="text-align: center; margin-top: 5px;"> <input type="button" value="«"/> <input type="button" value="Página 1 de 2"/> <input type="button" value="»"/> <span style="float: right;">Resultados encontrados 8</span> </p> </div> </div> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </div>	Prioridad	Entidad	Provincia	Municipio	Disponibilidad	<input type="radio"/>	1 UCI	La Habana	La Habana	40	<input type="radio"/>	2 MININT	La Habana	La Habana	10	<input type="radio"/>	3 Emp. Agropecuaria	Matanzas	Matanzas	6	<input type="radio"/>	4 Emp. Petróleo	Holguín	Holguín	1		Nombre y apellidos	Índice académico	Integralidad	Índice general		<input type="checkbox"/>	Abel Pérez	5.23	1.2	5.23		<input type="checkbox"/>	Laura Mola	5.01	1.2	5.01		<input type="checkbox"/>	Carlos Martínez	4.98	1.0	4.98		<input type="checkbox"/>	Pedro Díaz	4.50	0.96	4.50	
Prioridad	Entidad	Provincia	Municipio	Disponibilidad																																																				
<input type="radio"/>	1 UCI	La Habana	La Habana	40																																																				
<input type="radio"/>	2 MININT	La Habana	La Habana	10																																																				
<input type="radio"/>	3 Emp. Agropecuaria	Matanzas	Matanzas	6																																																				
<input type="radio"/>	4 Emp. Petróleo	Holguín	Holguín	1																																																				
	Nombre y apellidos	Índice académico	Integralidad	Índice general																																																				
<input type="checkbox"/>	Abel Pérez	5.23	1.2	5.23																																																				
<input type="checkbox"/>	Laura Mola	5.01	1.2	5.01																																																				
<input type="checkbox"/>	Carlos Martínez	4.98	1.0	4.98																																																				
<input type="checkbox"/>	Pedro Díaz	4.50	0.96	4.50																																																				
<p style="background-color: #d3d3d3; padding: 5px;"><b>Observaciones</b></p>	<ol style="list-style-type: none"> <li>1. Para realizar la acción el usuario tiene que estar autenticado en el sistema.</li> <li>2. Solo se mostrarán las entidades que tengan plazas disponibles.</li> <li>3. Un estudiante solo puede tener asignada una plaza laboral.</li> <li>4. Los estudiantes que ya estén asociados a una plaza no se deben mostrar en el</li> </ol>																																																							



		<p>listado (si se desea modificar la plaza a un estudiante debe ser por la opción de modificar plaza).</p> <ol style="list-style-type: none"><li>5. Cada vez que se asocie una plaza laboral a un estudiante se debe disminuir la disponibilidad de plazas laborales para esa entidad. Esto se aplica también para el caso en que se seleccione más de un estudiante.</li><li>6. Si el usuario presiona el botón Cancelar se muestra el mensaje de confirmación: “¿Está seguro de realizar la acción?”, si presiona Aceptar regresa al listado de Mostrar asignación de plazas. Si presiona el botón Cancelar se mantiene en la página sin realizar ninguna operación.</li><li>7. En caso que se seleccionen más estudiantes que la cantidad de plazas disponibles para una entidad, el sistema muestra el mensaje de error: “Ha seleccionado más estudiantes que la cantidad de plazas disponibles para esa entidad”, y no se realiza la operación.</li><li>8. La cantidad por página que se muestra en cada listado puede ser (5, 10, 15, 20), mostrando por defecto 5.</li><li>9. Interactúan con esta funcionalidad el administrador del sistema, el Asesor Técnico Docente y la rectora.</li></ol>
--	--	---

## 2.4 Descripción de la arquitectura y el diseño

En el desarrollo de la propuesta de solución se utilizará la arquitectura cliente-servidor, haciendo uso del estilo arquitectónico Modelo-Vista-Controlador, ya que el marco de trabajo empleado está basado en dicho estilo, por su flexibilidad y ventajas.

### 2.4.1 Arquitectura cliente-servidor

En una arquitectura cliente-servidor, una aplicación se modela como un conjunto de servicios proporcionados por servidores y un conjunto de clientes que usan estos servicios a través de una red; los clientes necesitan conocer qué servidores están disponibles, pero normalmente no conocen la existencia de otros clientes. (Sommerville, 2004). Una representación gráfica de este tipo de arquitectura sería la siguiente:

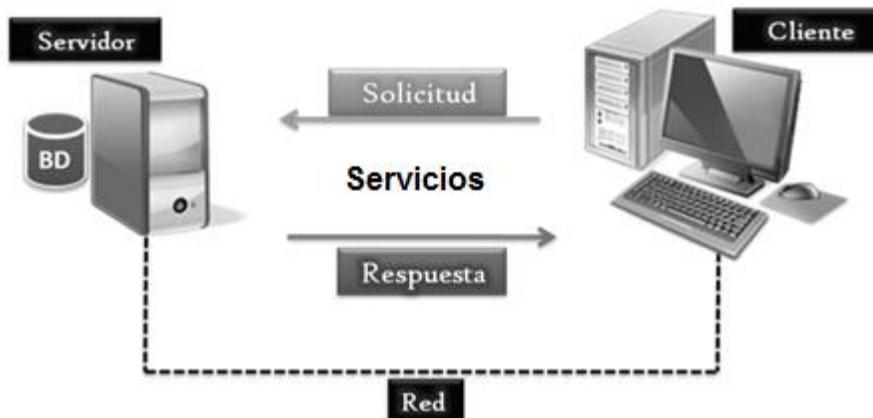


Figura 1: Arquitectura Cliente-Servidor.

## 2.4.2 Patrón de arquitectura

El patrón Modelo-Vista-Controlador (MVC) permite realizar la programación multicapa, separando el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

- **Modelo:** El modelo administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).
- **Vista:** Maneja la visualización de la información.
- **Controlador:** Interpreta las acciones del ratón y el teclado, informando al modelo y/o a la vista para que procesen la información según resulte apropiado.

En aplicaciones de Web, por otra parte, la separación entre la vista (el browser) y el controlador (los componentes del lado del servidor que manejan los requerimientos de HTTP) está mucho más taxativamente definida (Burbeck, 2010).

### Implementación del MVC que realiza GUUD

El patrón de arquitectura MVC se utiliza en el marco de trabajo GUUD sobre el cual se desarrolla la solución. El marco de trabajo cuenta en su interior con carpetas definidas para cada uno de los sistemas que forman parte del *software* a desarrollar. Cada uno de estos sistemas define carpetas para cada uno de sus módulos quedando demostrada la utilización del patrón de la manera siguiente:



- **Modelo:** en la carpeta *models* se encuentran los archivos que contienen una serie de funciones que permiten devolver, insertar y actualizar la información de la base de datos y por otro lado la carpeta *libraries* donde se encuentran los archivos que sirven de intermediarios entre el controlador y el modelo, además de ser usadas para obtener información de los restantes módulos. Esta última es una adaptación realizada para disminuir la cantidad de responsabilidades en el controlador.
- **Vista:** en la carpeta *views* serán almacenadas las páginas o fragmentos de páginas web que mostrarán la información al usuario.
- **Controlador:** en la carpeta *controllers* se encuentran los archivos que son los encargados de procesar la información y cargar cualquier recurso necesario para procesar la solicitud HTTP.

Esta separación permite construir y probar el modelo independientemente de la representación visual. En la **Figura 2** se ilustra la particularidad del MVC implementado por el GUUD.

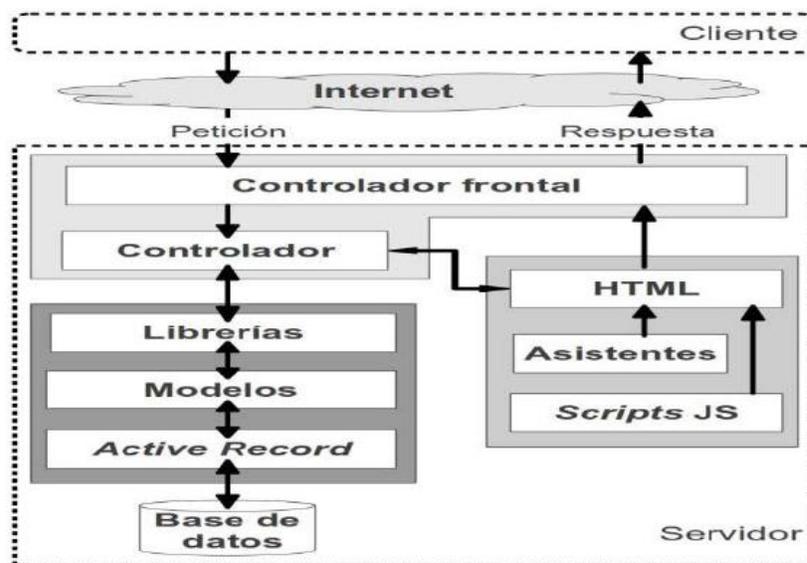


Figura 2: Funcionamiento del MVC en GUUD.

## 2.4.3 Modelo de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes *hardware* y *software* en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes *software* (procesos y objetos que se ejecutan en ellos). Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de



componentes *software*, objetos, procesos. En general un nodo será una unidad de computación de algún tipo (Object Managed Group, 2013). En la **Figura 3** se presenta el modelo de despliegue de la solución:

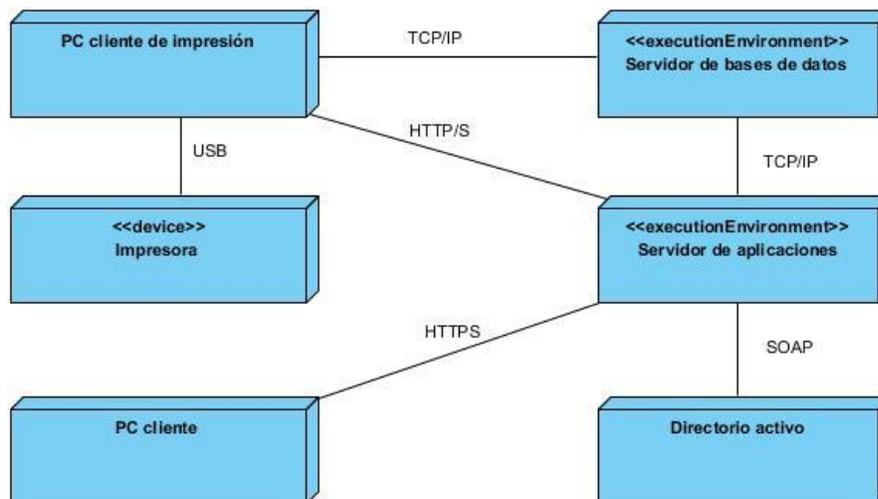


Figura 3: Diagrama de despliegue.

El diagrama de despliegue representado muestra la siguiente distribución:

- PC cliente: ordenador cliente capaz de conectarse al servidor de aplicaciones mediante el protocolo de comunicaciones HTTPS.
- PC cliente de impresión: ordenador cliente destinado a la impresión de boletas de ubicación laboral capaz de conectarse al servidor de aplicaciones mediante el protocolo de comunicaciones HTTP.
- Servidor de aplicaciones: ordenador en que se encuentra el servidor web Apache, este será el lugar en que se gestione todo el contenido de la aplicación. El mismo establecerá comunicación con los ordenadores clientes mediante protocolo HTTPS, con el servidor de base de datos por medio de la familia de protocolos TCP/IP y con el directorio activo de la universidad mediante el protocolo SOAP.
- Servidor de base de datos: ordenador en que se encuentra el gestor de base de datos PostgreSQL, el cual contiene las bases de datos del SGU y permite mantener persistente la información.
- Impresora: los informes que se generan en el sistema podrán ser llevados a formato duro mediante una impresora conectada a las PC clientes.



- HTTPS: es un protocolo de red basado en HTTP por lo que está orientado a transacciones sin estado, es decir, no guarda ninguna información sobre conexiones anteriores y sigue el esquema petición-respuesta entre un cliente y un servidor.
- HTTP: protocolo que rige la comunicación entre un cliente que utiliza un navegador web tal como Internet Explorer y un servidor web.
- TCP/IP: es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo computadoras personales, minicomputadoras y computadoras centrales sobre redes de área local y área extensa.
- USB (*Universal Serial Bus*): interfaz para la transmisión en serie de datos y distribución de energía.
- SOAP: protocolo estándar que define como dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

## 2.4.4 Patrones de diseño

Los patrones de diseño son abstracciones de alto nivel que documentan soluciones de diseño exitosas. Son fundamentales para reutilizar el diseño en el desarrollo orientado a objetos. Una descripción del patrón debería incluir un nombre del patrón, una descripción del problema y la solución, y una declaración de los resultados y compromisos al utilizar el patrón (Sommerville, 2004).

### Patrones Generales de Software para Asignación de Responsabilidades

Los patrones Generales de Software para Asignación de Responsabilidades (GRASP, por sus siglas en inglés General Responsibility Assignment Software Patterns) describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. Para el diseño de la propuesta de solución, se tuvo en cuenta 5 patrones GRASP: Experto en Información, Creador, Bajo acoplamiento, Alta cohesión y Controlador, con el propósito de lograr un sistema reusable y flexible.

A continuación se realiza una descripción del uso de estos patrones en la propuesta de solución:

- **Experto:** se evidencia en las clases librerías, que cuentan con la información necesaria para cumplir con la responsabilidad sobre los elementos del negocio. Su uso permite asignar a una clase la información necesaria para cumplir su responsabilidad, de esta forma, se asignan las mismas de forma adecuada y el módulo tiende a ser más fácil de entender, mantener y ampliar, pensando en una futura realización de componentes.



- **Creador:** se evidencia en el objeto *load* de la clase *loader* perteneciente a las clases controladoras, esta clase se encarga de iniciar los elementos del marco de trabajo, las librerías y modelos.
- **Controlador:** este patrón se refleja en las clases controladoras, que son las clases que se encargan de obtener datos y enviarlos a las librerías y vistas. El patrón controlador garantiza que los procesos sean manejados por la capa controladora y no por la capa de presentación.
- **Alta Cohesión y Bajo Acoplamiento:** la propia implementación del CodeIgniter contiene estos patrones nivelados, permitiendo el uso de los componentes de forma individual, evidenciando de esta forma el bajo acoplamiento, así como la dependencia entre los componentes o la alta cohesión (García Vidal, 2012).

## Patrones Banda de los Cuatros

Los patrones conocidos como Banda de los Cuatros (GoF, por sus siglas en inglés Gang of Four) son descripciones de la comunicación entre objetos y clases adaptadas a la resolución de problemas generales de diseño en un contexto particular. Los mismos se dividen en tres categorías diferentes:

- **Creacionales:** tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es abstraer el proceso de instanciación y ocultar los detalles de como los objetos son creados o inicializados.
- **Estructurales:** describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples o compuestos.
- **Comportamiento:** ayudan a definir la comunicación e interacción entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos (Gamma, y otros, 1994).

A continuación se realiza la descripción del uso de los patrones GoF empleados en la propuesta de solución:

- **Instancia única:** garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Este patrón se refleja en las clases controladoras que son instancias únicas y en la clase IoC para la interacción entre módulos. Los patrones de comportamiento estudian las relaciones de llamadas entre los diferentes objetos, normalmente ligados con la dimensión temporal.



- **Mediador:** define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto. Se evidencia en las librerías, las cuales son mediadoras entre las clases controladoras y los modelos.
- **Observador:** define una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Se refleja en la clase *loader* que es el objeto *load* de las clases controladoras, cuya función es cargar los elementos del marco de trabajo dígame librerías, modelos y se encarga de actualizar la controladora instanciada (García Vidal, 2012).

## 2.4.5 Patrones de diseño de base de datos

Los patrones de diseño de base de datos permiten al usuario crear una base de datos más fortalecida ya que constituyen una guía que especifica cómo debe ser la misma. El diseño y construcción de una base de datos requiere del mayor esfuerzo y análisis posible, ya que la calidad con la que esta se obtenga determinará su comportamiento futuro (Entorno Virtual de Aprendizaje. EVA, 2012). A continuación se describen cada uno de los patrones de base de datos utilizados para desarrollar la propuesta de solución.

- **Llaves subrogadas:** este patrón es muy utilizado pues facilita la interacción con la base de datos en un futuro. El mismo plantea que se genere una llave primaria única para cada entidad, en vez de usar un atributo identificador en el contexto dado. Normalmente se usan números enteros en columnas *identity* o GUID (*Global Unique Identifier*) que está demostrado que no se repiten o con una probabilidad extremadamente baja. Esto permite que las tablas sean más fáciles de consultar a partir del identificador, pues todos tienen el mismo tipo en cada una de las tablas (Entorno Virtual de Aprendizaje. EVA, 2012).

Mediante este patrón, fueron generados los identificadores de todas las tablas pertenecientes al modelo de datos que se muestra en la **Figura 10**.

- **Entidad-Atributo-Valor:** es la representación de un modelo flexible donde se pueden representar objetos con sus atributos. Es un acercamiento al modelo orientado a objeto representado en el modelo relacional, donde la entidad *Class* representa las clases, la entidad *Attribute* representa los atributos de las clases, por su parte la entidad *Object* representa las instancias de las clases, mientras que la entidad *Value* representa los valores de cada atributo para cada objeto dado (Entorno Virtual de Aprendizaje. EVA, 2012), como se muestra en la **Figura 4**.



El modelo de datos propuesto en el acápite 2.6 presenta dicho patrón entre las tablas: tb\_dcomición -> tb\_rpersona\_comision -> tb\_dpersona.

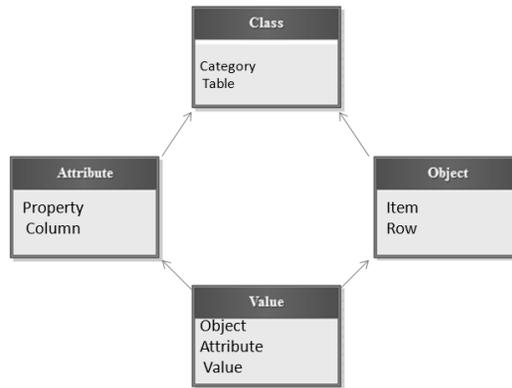


Figura 4: Caracterización del modelo entidad-atributo-valor.

- **Árboles fuertemente codificados:** a cada nivel del árbol se le asocia una entidad. Normalmente constituyen relaciones de uno a muchos (n). Se utiliza para representar jerarquías donde es conocida la estructura y es importante representar la correspondencia, por ejemplo las estructuras organizacionales. Además admite tantos niveles como requiera la jerarquía que se vaya a representar (Entorno Virtual de Aprendizaje. EVA, 2012), como se muestra en la **Figura 5**.

El modelo de datos propuesto en el acápite 2.6 presenta dicho patrón entre las tablas: tb\_norganismo -> tb\_dempresa -> tb\_ddemanda.

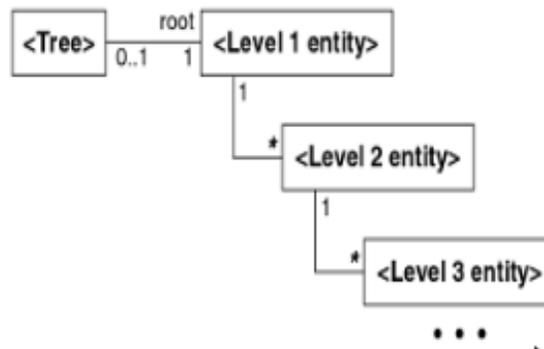


Figura 5: Árbol fuertemente codificado.



## 2.5 Componentes visuales del sistema

El diseño del módulo Ubicación Laboral fue realizado siguiendo las pautas del diseño establecidas en el artefacto “*Vista de Presentación*” del expediente del proyecto Egreso, posibilitando la uniformidad en la estructura del sistema.

### 2.5.1 Vistas principales

#### ❖ Vista de Presentación

La vista de presentación es la primera vista que se le muestra al usuario, a través de la cual podrá autenticarse y acceder al sistema. La misma se describe a continuación.

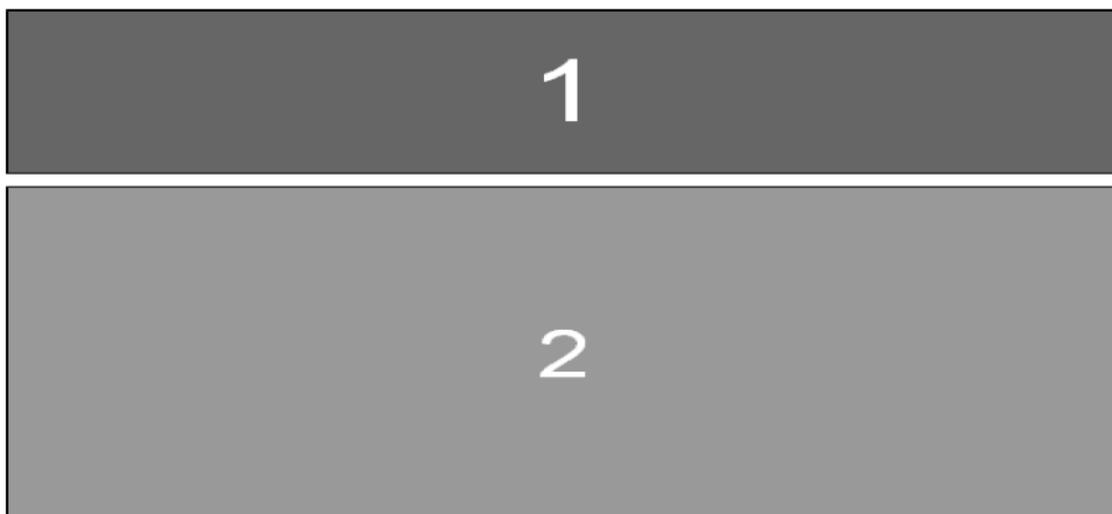


Figura 6: Áreas de la vista de presentación.

1- Área de identificación o cabezal.

2- Área de entrada de datos.

#### ❖ Vista de escritorio

La vista de escritorio se muestra luego de la autenticación del usuario, permite seleccionar el subprocesso horizontal, módulo o servicio al que desea acceder y tenga los permisos requeridos. La cual se describe a continuación.

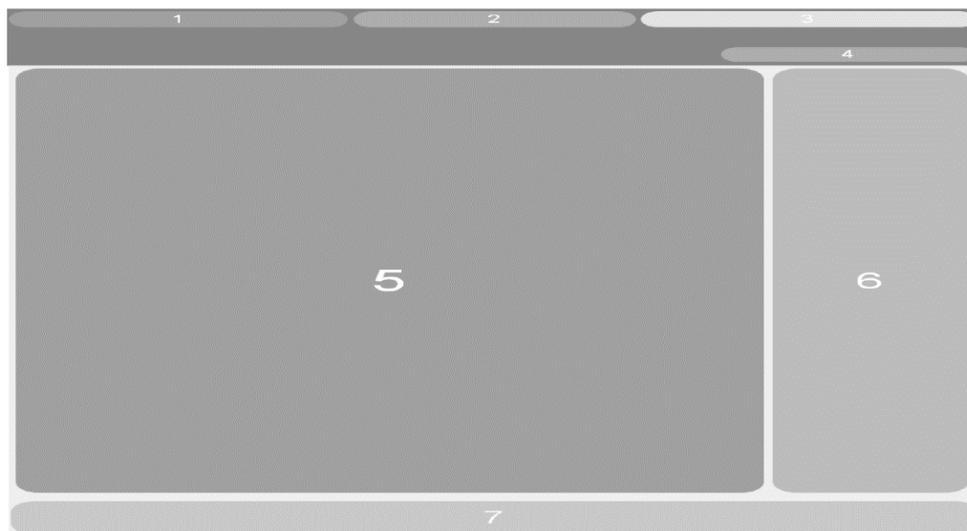


Figura 7: Áreas de la vista de escritorio.

- |                                       |                                    |
|---------------------------------------|------------------------------------|
| 1- Área de subprocessos horizontales. | 5- Área de líneas de procesos.     |
| 2- Área del nombre de la aplicación.  | 6- Área de servicios horizontales. |
| 3- Área del buscador.                 | 7- Área de pie de página.          |
| 4- Área del nombre del usuario.       |                                    |

## ❖ Vista de procesos

La vista de gestión de procesos permite el acceso a los módulos del sistema seleccionado por el usuario y las funcionalidades que poseen. La cual se describe a continuación.

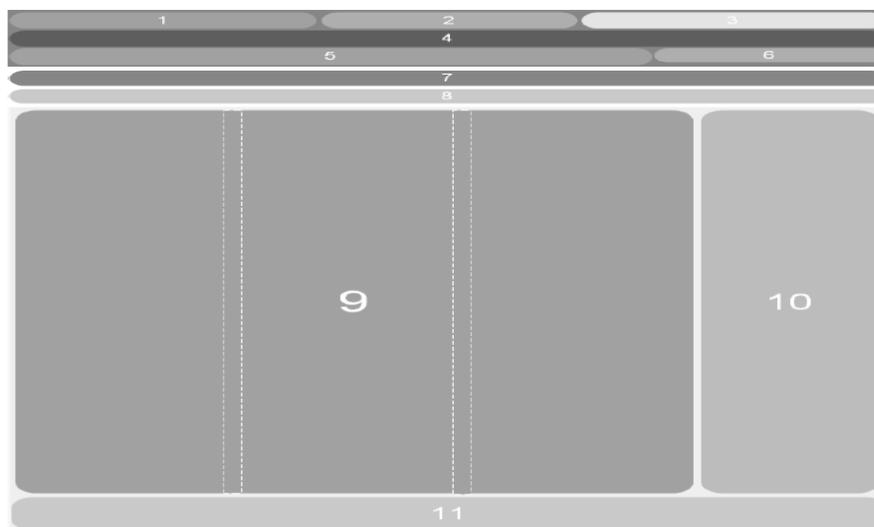


Figura 8: Áreas de la vista de gestión de procesos.



- |                                      |                               |
|--------------------------------------|-------------------------------|
| 1- Área de subprocesos horizontales. | 7- Área de traza.             |
| 2- Área del nombre de la aplicación. | 8- Área de línea de progreso. |
| 3- Área del buscador.                | 9- Área de contexto.          |
| 4- Área de líneas de procesos.       | 10- Área del menú de módulos. |
| 5- Área de módulos.                  | 11- Área de pie de página.    |
| 6- Área de nombre de usuario.        |                               |

## 2.6 Mapa de Navegación

El módulo Ubicación Laboral está compuesto por cinco agrupaciones funcionales, la primera se ocupa de gestionar los elementos necesarios para llevar a cabo la ubicación laboral, la segunda de la gestión de la Ubicación Laboral de los estudiantes, la tercera de la gestión de los tipos de comisión, los elementos involucrados en la asignación de responsabilidades de las personas asociadas a las comisiones, los organismos, las entidades priorizadas y de las prioridades de las entidades no priorizadas, la cuarta de la gestión de los grupos de certificación, necesario para la impresión de las boletas de ubicación laboral y la quinta de los reportes necesarios para el Asesor Técnico Docente. Las funcionalidades del módulo se agruparon según las opciones o procesos que gestionan, como se muestra en la siguiente figura.



Figura 9: Mapa de navegación.



## 2.7 Modelo de datos

Un modelo de datos es la descripción de una base de datos. Típicamente un modelo de datos permite describir las estructuras de la base de datos, la forma en que se relacionan, las restricciones de integridad entre otros. Es factible pensar que un modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí (Martin's, 1990).

En la siguiente figura se muestra una representación del modelo físico de la base de datos, con un total de 13 clases.

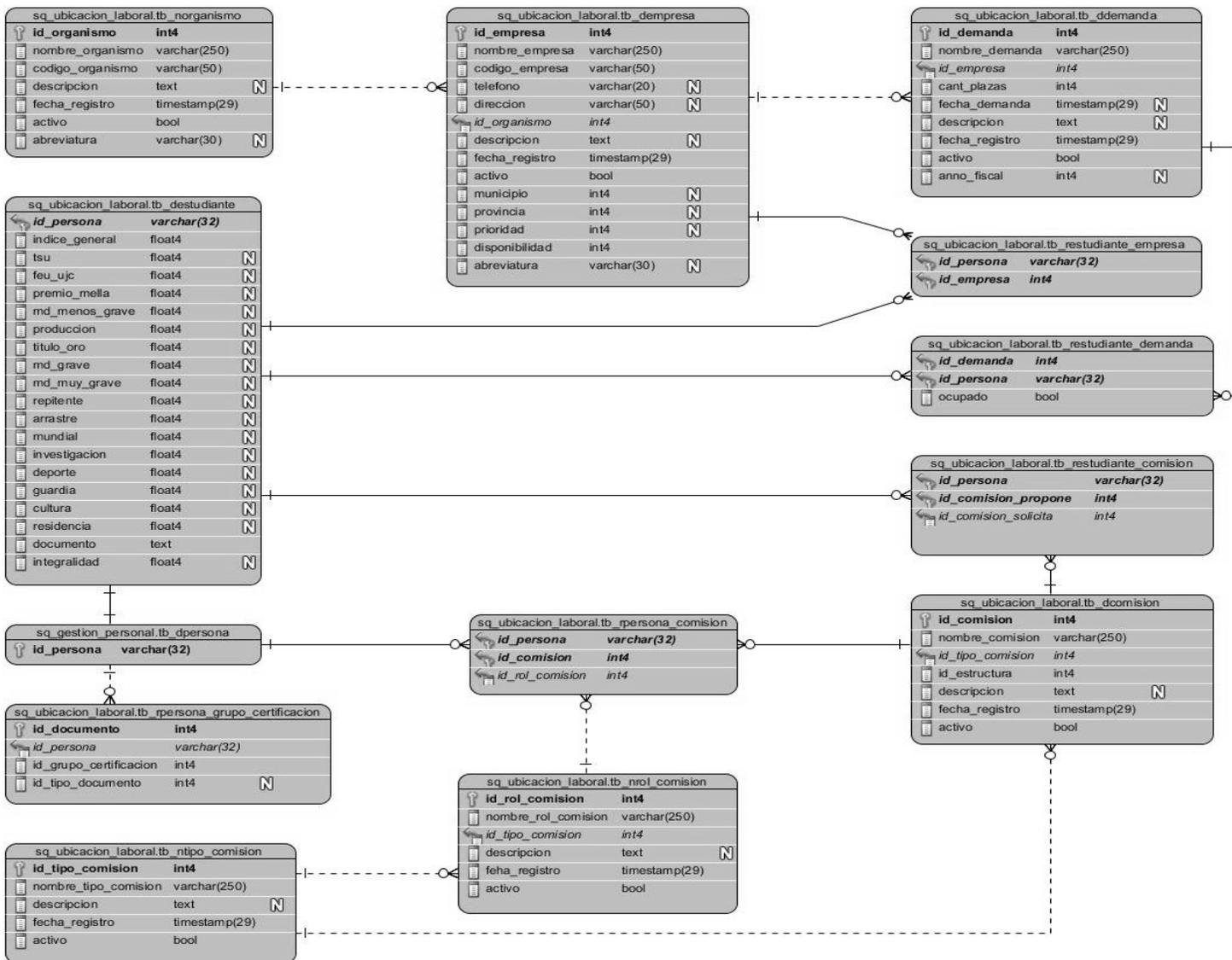


Figura 10: Modelo físico de datos.



## 2.8 Conclusiones del capítulo

En este capítulo se describieron las actividades que integran el proceso de Ubicación Laboral y las características fundamentales de la propuesta de solución, lo que propició un mejor entendimiento para definir dicha propuesta; la que está encaminada a satisfacer las necesidades de la Oficina de Ubicación Laboral de la Vicerrectoría de Producción. El estudio de las técnicas de obtención de requisitos permitió la definición de los requisitos funcionales y no funcionales. Se describió la arquitectura, el patrón MVC, los patrones de diseño y los de base de datos, así como la obtención del modelo de datos y la distribución física de la propuesta de solución. Al terminar este capítulo se puede concluir que están creadas las condiciones necesarias para el desarrollo y validación del módulo de Ubicación Laboral.



## **CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN**

En este capítulo se describen los paradigmas de programación y los estándares de codificación empleados en el desarrollo de la propuesta de solución. Además se describen las técnicas de validación de requisitos y las pruebas de *software* aplicadas, así como sus resultados, lo que permite verificar el cumplimiento de los requisitos definidos y garantizar el correcto funcionamiento de la solución desarrollada.

### **3.1 Paradigmas de programación**

Un paradigma de programación es una propuesta tecnológica que es adoptada por una comunidad de programadores cuyo núcleo central es incuestionable en cuanto a que únicamente trata de resolver uno o varios problemas claramente delimitados. Es un estilo de programación empleado. La resolución de estos problemas debe suponer consecuentemente un avance significativo en al menos un parámetro que afecte a la ingeniería de software (Bonanata, 2000).

#### **Programación Orientada a Objetos**

La programación orientada a objetos (POO) es una forma de estructurar un programa sobre la base de objetos. Cada elemento o componente en un programa que se base en esta técnica es concebido como un objeto que tiene propiedades y métodos. La ejecución de este tipo programa depende pura y exclusivamente de una interacción de los objetos que lo componen (Ing. Maximiliano Bonanata, 2000).

En GUUD la POO está presente en la mayoría de sus componentes, muestra de ello son las librerías, que son clases preparadas para realizar tareas típicas en el desarrollo de páginas web. Las mismas implementan clases de POO para resolver problemas muchas veces similares a los que resuelven los *Helpers*, pero son un poco más especializadas en temas particulares.

#### **Programación Orientada a Eventos**

La programación orientada a eventos se fundamenta principalmente en la ejecución de funciones y procedimientos dada la ocurrencia de distintos eventos durante la ejecución de un programa (Bonanata, 2000).

La programación orientada a eventos se evidencia en el marco de trabajo GUUD a través de los ficheros JavaScript de cada vista, los cuales son los encargados de administrar los componentes visuales y validar



las acciones de los usuarios mediante la ejecución de funciones y procedimientos dada la ocurrencia de distintos eventos.

## 3.2 Estándar de codificación

Los estándares de codificación representan un estilo de programación homogénea en un proyecto. Son de vital importancia durante la etapa de construcción del *software* debido a que permiten que todo el personal del proyecto pueda entender de forma fácil el código, garantizándose la organización y estructura del código fuente.

Para el desarrollo del módulo se utilizaron los estándares de codificación establecidos por el Grupo de Arquitectura del Departamento de Desarrollo de la Dirección de Informatización, con el propósito de estandarizar las nomenclaturas en la implementación del sistema.

### 3.2.1 Identación, llaves de apertura y cierre, y tamaño de las líneas

Se debe usar la indentación sin tabulaciones, con un equivalente a 4 espacios, para mantener integridad en las revisiones svn<sup>8</sup>. El uso de las llaves “{}” será en una nueva línea. La longitud de las líneas de código es aproximadamente de 75-80 caracteres, para mantener la legibilidad del código.

Ejemplo:

```
public function obtenerNombreEstudianteDadoIdPersona ($id)
{
    $nombre = $this->_ci->tb_destudiante_md1->obtenerNombreEstudianteDadoIdPersona ($id);
    return $nombre;
}
```

Figura 11: Identación, llaves de apertura y cierre.

### 3.2.2 Convención de nomenclatura

**Variables:** se rigen por la nomenclatura *camelCase*, siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

Ejemplo:

---

<sup>8</sup> Sistema de control de versiones.



```
$integralidad=($valorSuma-$valorAgravante)/100;  
$valorIndice=$integralidad+$promedio;
```

Figura 12: Variables.

**Clases:** siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con el carácter subrayado “\_” y el resto en minúscula.

Ejemplo:

```
class Ubicacion_laboral extends MY_Controller  
{  
    public function __construct() {  
        parent::__construct();  
        $this->load->library('ubicacion_laboral_lib');  
    }  
}
```

Figura 13: Clases.

**Funciones:** se rigen por la nomenclatura *camelCase*, siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa.

Ejemplo:

```
public function obtenerMunicipioDadoIdProvincia($idProvincia)  
{  
    $municipio = $this->_ci->tb_dempresa_mdl->obtenerMunicipio($idProvincia);  
    return $municipio;  
}
```

Figura 14: Funciones.

**Ficheros:** siempre se escriben en minúscula y en caso de nombres compuestos se usa el carácter subrayado “\_”.

**Vistas:** intuitivo y relacionado con el formulario y/o vista que representa.

**Modelos:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo `_mdl`.

**Librerías:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo `_lib`.

**Controladoras:** con el mismo nombre de la clase que representa.

### 3.2.3 Documentación



Todos los archivos deben de tener la documentación asociada al mismo. Para esto debe de cumplir con el siguiente bloque al principio de cada clase.

Ejemplo:

```
/**
 * Librería asignar_prioridad_entidad_lib para el módulo Ubicación Laboral
 *
 * @author fmorales
 */
```

Figura 15: Documentación.

### 3.2.4 Estructuras de control

Se incluye un espacio entre las estructuras de control (*if*, *for*, *foreach*, *while*, *switch*) y los paréntesis. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos.

Ejemplo:

```
if ($this->Indices_estudiante_lib->modificarIndicesEstudiante($datos) != FALSE)
{
    $this->message('SYS002');
}
else
{
    throw new Exception_Error('SYS006');
}
```

Figura 16: Estructuras de control.

### 3.3 Validación de la propuesta de solución

En el proceso de construcción de un *software* la calidad no se puede probar, esta se va construyendo a lo largo del desarrollo del mismo. Cuanto más errores se encuentren en las revisiones, más se podrá asegurar la calidad de la solución desarrollada.

En la investigación al utilizar un proceso de desarrollo con enfoque ágil basado en el nivel 2 de CMMI se deben aplicar validaciones a los requisitos y realizar pruebas durante el ciclo de vida del producto. En los siguientes epígrafes se podrán encontrar las validaciones efectuadas a los requisitos y la estrategia de pruebas a utilizar.



## 3.4 Validación de los requisitos

La validación de requisitos es el proceso por el cual se determina si los requisitos relevados son consistentes con las necesidades del cliente. Además examina las especificaciones para asegurar que los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos (Pressman, 2005).

### 3.4.1. Criterios de validación de requisitos del cliente

En el proceso de desarrollo se define el documento para la validación de los requisitos del cliente "*Criterios para validar requerimientos del cliente*" con el que se pudo conocer si los requisitos funcionales y no funcionales quedaron aprobados después de evaluar todos los criterios definidos. Los criterios de validación aparecen recogidos en el **Anexo 4: Criterios para validar requerimientos del cliente**.

#### Resultado de aplicar los criterios de validación

Con la aplicación de los criterios para validar requisitos del cliente, se logró obtener el 100 % de los requisitos aprobados como se refleja en el artefacto "*Criterios para validar requisitos del cliente*" del expediente del proyecto Egreso.

### 3.4.2. Técnicas de validación de los requisitos

Las siguientes técnicas empleadas para la validación de los requisitos fueron definidas por Ian Sommerville en el libro Ingeniería de *software*, y permiten que el proceso de validación de la solución tenga una mejor calidad:

- ❖ **Revisiones de requisitos:** se realizaron reuniones con el cliente para la lectura, revisión y corrección de la definición de los requisitos, de este modo se pudo obtener los errores presentes en la especificación de los mismos y verificar la interpretación de la información.
- ❖ **Construcción de prototipos:** este enfoque se realizó con el objetivo de facilitar a los usuarios interesados en el producto una imagen de lo que sería el futuro sistema, a partir de los requisitos recogidos en la especificación, entregando una versión del producto final. A través de los prototipos se pudo comprobar que se estaba construyendo el sistema correcto, que la aplicación producía las salidas correctas y cumplía con las necesidades y requisitos del usuario.



- ❖ **Generación de casos de prueba:** la realización de casos de prueba posibilitó la verificación del cumplimiento de los requisitos funcionales y se pudo comprobar que los mismos tenían la eficiencia requerida.

## Resultado de la validación de requisitos

Como resultado del proceso de validación de requisitos se identificaron inconsistencias en las especificaciones, tales como:

- ❖ Falta de concordancia entre la complejidad de la especificación y la registrada en el documento “*Evaluación de Requisitos*”
- ❖ Componentes innecesarios en los prototipos de interfaz.
- ❖ Descripciones poco detalladas de algunos de los requisitos.
- ❖ Errores ortográficos.

## 3.5 Estrategias de pruebas

La estrategia de prueba del *software* describe el enfoque y los objetivos generales de las actividades de prueba. Esta debe incluir pruebas de bajo nivel que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente (Pressman, 2005).

La estrategia que se ha de seguir al evaluar dinámicamente un sistema debe permitir comenzar por los componentes más simples y pequeños e ir avanzando progresivamente hasta probar todo el *software* en su conjunto. Para los efectos de la investigación se define una estrategia basada en las técnicas y métodos propuestos por *Roger S. Pressman* en la cual se propone realizar:

1. Prueba de unidad.
2. Prueba de integración.
3. Prueba de validación.
4. Prueba del sistema.

A continuación se procede a la descripción y realización de cada una de las pruebas antes mencionadas.

### 3.5.1 Prueba de unidad

La prueba de unidad centra el proceso de verificación en la menor unidad del diseño del software: el componente software o módulo (Pressman, 2005).



Se prueba la interfaz del módulo para asegurar que la información fluye de forma adecuada hacia y desde la unidad de programa que está siendo probada. Además son examinadas las estructuras de datos locales para asegurar que los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución del algoritmo. También se prueban las condiciones límites para asegurar que el módulo funciona correctamente en los límites establecidos como restricciones de procesamiento. Se ejercitan todos los caminos independientes (caminos básicos) de la estructura de control con el fin de asegurar que todas las sentencias del módulo se ejecutan al menos una vez.

## **Método de Caja Blanca**

La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba (Pressman, 2005).

## **Técnica del camino básico**

La técnica utilizada en la investigación para realizar el método de caja blanca es el camino básico, que permite obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Para obtener dicho conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática.

Para la realización de las pruebas de caja blanca y con el objetivo de verificar el resultado real de la prueba en cada uno de los caminos, se empleó un mecanismo para la automatización de pruebas unitarias que posee CodeIgniter, el cual es parte estructural del marco de trabajo y que se describe a continuación:

CodeIgniter posee una librería o clase especializada en la ejecución de pruebas estructurales. Aunque es bastante sencilla, pues cuenta con una sola función de evaluación y dos funciones de resultados, permite determinar con certeza si un código específico produce el tipo de dato y resultado esperado. Para realizar una prueba utilizando dicha librería es necesario suministrar el código a probar y un resultado esperado de la siguiente forma:

```
$this->unit->run (código, resultado esperado, 'nombre de prueba');
```

Donde el primer parámetro es el segmento de código que se desea probar, el segundo es lo que se espera que devuelva la evaluación del código (puede ser un tipo de dato o un valor literal) y el tercero es un nombre opcional que se le puede dar a la prueba.



Los tipos de datos posibles son: "is\_string", "is\_bool", "is\_true", "is\_false", "is\_int", "is\_numeric", "is\_float", "is\_double", "is\_array", "is\_null". En la **Tabla 4** se muestra un ejemplo de un caso de prueba de caja blanca realizado a la solución.

Tabla 4: Casos de pruebas de caja blanca del método obtenerCantidadPersonas.

Prueba estructural de caja blanca		Código caso de prueba: DIN_Egre_UL_OF
Probador: Froilán Morales Medina		
Código al que se aplica:		
<pre>public function obtenerCantidadPersonas(\$datos_buscador = null) {     if (\$datos_buscador-&gt;cargo != ''    \$datos_buscador-&gt;text_buscar!= '')         return count(\$this-&gt;obtenerPersonas(null, null, null, null, \$datos_buscador));     else         return 0; }</pre>		
<b>Complejidad ciclomática:</b> $V(G) = A - N + 2$ $V(G) = 4 - 4 + 2 = 2$ <b>Caminos independientes:</b> 1-2-4 1-3-4	<b>Representación del grafo:</b> <pre>graph TD     1((1)) --&gt; 2((2))     1((1)) --&gt; 3((3))     2((2)) --&gt; 4((4))     3((3)) --&gt; 4((4))</pre>	
Caso de prueba para el camino básico 1		
<b>Descripción:</b> los datos de entrada serán los atributos de las personas.		
<b>Condición de ejecución:</b> los datos de entrada están vacíos.		
Procedimiento prueba automatizada		
<b>Datos de entrada:</b>	Un objeto con los datos de las personas vacío.	
<b>Tipo de dato esperado:</b>	is_numeric.	
<b>Función de evaluación:</b>		
<pre>\$nombrePrueba="Obtener cantidad persona"; \$resultadoEsperado="is_numeric"; echo \$this-&gt;_ci-&gt;unit-&gt;run(\$this-&gt;obtenerCantidadPersonas(\$datos_buscador), \$resultadoEsperado, \$nombrePrueba);</pre>		
<b>Evaluación del caso de prueba:</b>		



Test Name	Obtener cantidad persona
Test Datatype	Integer
Expected Datatype	numeric
Result	Passed
File Name	C:\wamp\www\nucleo\base\application\egreso\ubicacion_laboral\libraries\comision_lib.php
Line Number	26
<b>Caso de prueba para el camino básico 2</b>	
<b>Descripción:</b> los datos de entrada serán los atributos de las personas.	
<b>Condición de ejecución:</b> los datos de entrada están listos para realizar la búsqueda.	
<b>Procedimiento prueba automatizada</b>	
<b>Datos de entrada:</b>	objeto con los datos de las personas.
<b>Tipo de dato esperado:</b>	is_numeric.
<b>Función de evaluación:</b>	
<pre> \$nombrePrueba="Obtener cantidad persona"; \$resultadoEsperado="is_numeric"; echo \$this-&gt;_ci-&gt;unit-&gt;run(\$this-&gt;obtenerCantidadPersonas (\$datos_buscador) , \$resultadoEsperado, \$nombrePrueba);                     </pre>	
<b>Evaluación del caso de prueba:</b>	
Test Name	Obtener cantidad persona
Test Datatype	Integer
Expected Datatype	numeric
Result	Passed
File Name	C:\wamp\www\nucleo\base\application\egreso\ubicacion_laboral\libraries\comision_lib.php
Line Number	26
<b>Resultado final de la prueba:</b> satisfactoria en su totalidad.	

En el **Anexo 5: Casos de prueba de caja blanca** se muestra el resto de los casos de prueba de caja blanca realizados a la solución.

## Resultado de las pruebas de caja blanca

Al aplicar la prueba de caja blanca se alcanzaron los siguientes resultados en cada una de las iteraciones, como se muestra en la **Tabla 5**. Las no conformidades encontradas fueron resueltas en su totalidad.

Tabla 5: No conformidades por iteraciones caja blanca.

Iteraciones	N° No conformidades	Asociadas a
1ra	7	Errores de validación, entradas y salidas incorrectas.
2da	2	Errores de validación.
3ra	0	-



## 3.5.2 Prueba de integración

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es coger los módulos probados mediante la prueba de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño (Pressman, 2005).

Existen dos tipos de integración: no incremental e incremental. En el primer caso se combinan todos los módulos por anticipado y se prueba todo el programa en conjunto, normalmente se encuentra un gran conjunto de errores. La corrección se hace difícil, puesto que es complicado aislar las causas al tener delante el programa entero en toda su extensión. En el segundo caso el programa se construye y se prueba en pequeños segmentos en los que los errores son más fáciles de aislar y de corregir, es más probable que se puedan probar completamente las interfaces. Por esta razón se escogió el enfoque incremental para la realización de las pruebas de integración de la solución y se utilizó la técnica de casos de prueba de caja negra.

A continuación se muestra en la **Tabla 6** un ejemplo de caso de prueba de integración. En este se integran las funcionalidades Mostrar facultad y Obtener facultad grupo docente de los módulos Ubicación Laboral y Estructura y Composición. Los demás casos de prueba generados aparecen documentados en el **Anexo 6: Pruebas de integración**.

Tabla 6: Prueba de integración entre las funcionalidades Mostrar facultad y Obtener facultad.

N° de caso de prueba: Int_UL_1	
<b>Funcionalidad a integrar:</b>	Mostrar facultad.
<b>Módulo al que se integra:</b>	Estructura y Composición.
<b>Condiciones de ejecución:</b>	El usuario debe estar autenticado en el sistema. El módulo Estructura y Composición haya introducido los datos en la base de datos central y exista conexión con la misma.
<b>Descripción de la prueba:</b>	Comprobar que el módulo de Ubicación Laboral sea capaz de mostrar las facultades con información gestionada por el módulo Estructura y Composición.
<b>Entrada/Pasos de ejecución:</b>	El usuario se dirige a la agrupación funcional Ubicación Laboral y accede a la funcionalidad indicadores de integralidad.
<b>Resultado esperado:</b>	Se muestra las facultades de la universidad.



<b>Evaluación:</b>	Prueba satisfactoria.
--------------------	-----------------------

## Resultado de las pruebas de integración incremental

Los casos de prueba se fueron aplicando de forma incremental, donde se probó la integración con cada uno de los módulos a los que se integra la solución desarrollada. Luego de haber aplicado los casos de prueba se pudo identificar que en el 100 % de estos el resultado fue satisfactorio.

### 3.5.3 Prueba de validación

Tras la culminación de la prueba de integración, el software está completamente ensamblado como un paquete, se han encontrado y corregido los errores de interfaz, por lo tanto se puede comenzar una serie final de pruebas del software, entre la que figura la prueba de validación.

La validación puede definirse de muchas formas, pero una simple definición es que la validación se consigue cuando el software funciona de acuerdo con las expectativas razonables del cliente (Pressman, 2005). Para validar la aceptación de la solución informática por parte del cliente se aplicó la Prueba alfa.

#### Prueba alfa

Es virtualmente imposible que un desarrollador de software pueda prever como utilizará el usuario realmente el programa. Se pueden malinterpretar las instrucciones de uso, se pueden utilizar habitualmente extrañas combinaciones de datos, y una salida que puede parecer clara para el responsable de las pruebas y puede ser ininteligible para el usuario (Pressman, 2005). Cuando se construye software a medida para un cliente, se llevan a cabo una serie de pruebas de aceptación para permitir que el cliente valide todos los requisitos. Estas pruebas las realiza el usuario final en lugar del responsable del desarrollo del sistema, una prueba de aceptación puede ir desde un informal paso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas.

#### Resultado de la prueba alfa

Para el desarrollo de esta prueba se tomó como referencia el artefacto “Especificación de Requisito de Software”. Además el cliente usó el software de forma natural con el desarrollador como observador, registrando los posibles errores y los problemas de uso en un entorno controlado. Al finalizar las pruebas se comprobó que el sistema cubre satisfactoriamente los requisitos acordados. Por tanto, el desarrollo de esta prueba fue satisfactorio, quedando como constancia el acta de aceptación del cliente.



## 3.5.4 Prueba del sistema

La prueba del sistema según (Pressman, 2005), está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas. Para la realización de las pruebas del sistema se utilizó las pruebas de rendimiento.

### Prueba de rendimiento

Refiriéndose a las pruebas de rendimiento (Pressman, 2005) plantea que están diseñada para probar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado. La prueba de rendimiento se realiza durante todos los pasos del proceso de la prueba y a menudo, van emparejadas con las pruebas de resistencia y, frecuentemente, requieren instrumentación tanto de *software* como de *hardware*.

### Resultados de las pruebas de rendimiento

Para llevar a cabo las pruebas de rendimiento se utilizó la herramienta JMeter descrita en el acápite 1.5.8. A continuación se muestran un ejemplo de los resultados obtenidos con la misma.



Label	# Muestras	Media	Mediana	Línea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/sgu/base/ubic...	100	3407	3574	5387	320	8370	0,00%	1,9/sec	,4
/sgu/base/ubic...	100	3172	3377	4574	295	8666	0,00%	1,8/sec	1,6
/sgu/base/ubic...	100	4251	4231	5542	680	9017	0,00%	1,7/sec	19,3
/sgu/base/ubic...	100	4529	4410	6467	796	9172	0,00%	1,5/sec	1,2
/sgu/base/ubic...	100	4128	4104	5543	939	8342	0,00%	1,5/sec	4,8
/sgu/base/ubic...	100	4130	4224	5630	580	8372	0,00%	1,5/sec	,1
TOTAL	600	3936	4033	5543	295	9172	0,00%	8,8/sec	24,5

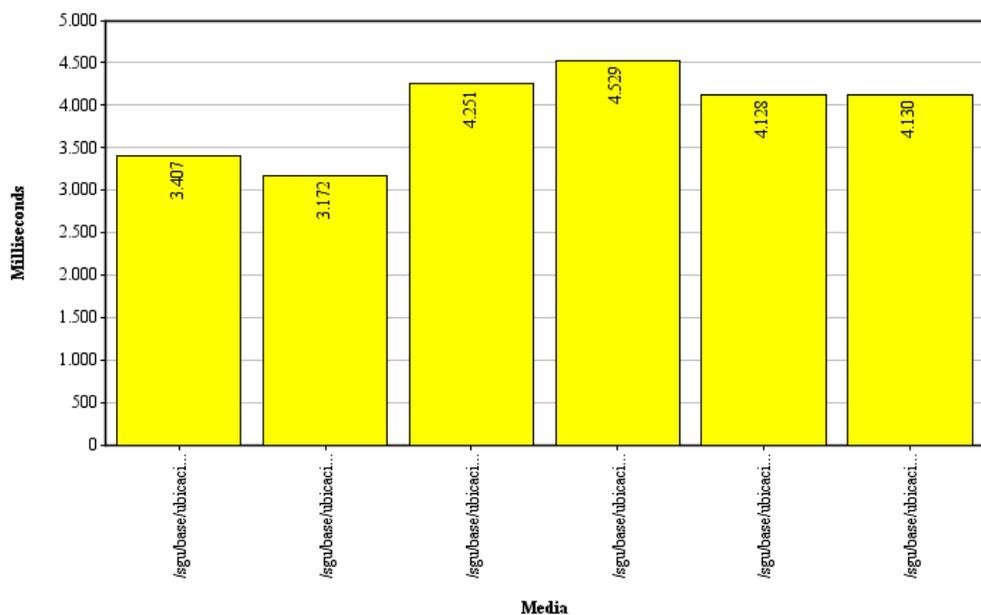


Figura 17: Resultados de las pruebas de rendimiento para 100 usuarios.

**Columna Mínimo (Mín.):** indica el mínimo de tiempo de ejecución invertido para una petición con  $n$  (columna *Muestras*) usuarios haciendo peticiones de manera concurrente.

**Columna Máximo (Máx.):** indica el máximo de tiempo de ejecución invertido para una petición con  $n$  (columna *Muestras*) usuarios haciendo peticiones de manera concurrente.

**Media:** representa el tiempo de ejecución promedio de una petición con  $n$  usuarios.

**Mediana:** significa que el 50% de las peticiones realizadas por  $n$  usuarios tardaron menos del valor reflejado.

**Línea de 90%:** tiempo de respuesta en el que se encuentra el 90% de los resultados.

**Error:** indica la relación entre el total de peticiones y el número de peticiones que originaron errores.

**Rendimiento (Pet./seg):** hace referencia al número de peticiones que el servidor puede procesar en un segundo.



**Rendimiento (Kb./seg):** hace referencia a la cantidad de datos que el servidor puede procesar en un segundo.

En la **Figura 17** se puede observar que el tiempo medio de respuesta del sistema es de 3.936 segundos para 100 usuarios conectados de forma concurrente, realizándose la prueba en una computadora con las siguientes prestaciones: Procesador Celeron M a 1.47 GHz de CPU, 2 GB de RAM y 120 GB de disco duro. Para todas las muestras de usuarios la ocurrencia de errores se mantiene en 0.0%, lo que significa que todas las peticiones hechas se ejecutan satisfactoriamente. Considerando que la cantidad máxima de usuarios conectados al módulo de Ubicación Laboral de forma concurrente no excederá los 10, se decidió realizar las pruebas de rendimiento para un máximo de 100 usuarios conectados de forma concurrente; ya que esta cantidad la supera en 9 veces.

En el **Anexo 7: Pruebas de rendimiento** se muestran las restantes pruebas de rendimiento realizadas al sistema.

## **Pruebas de caja negra**

Según (Pressman, 2005) las pruebas de caja negra son diseñadas para validar los requisitos funcionales sin fijarse en el funcionamiento interno de un programa. Las técnicas de prueba de caja negra se centran en el ámbito de información de un programa, de forma que se proporcione una cobertura completa de prueba.

## **Método de particiones equivalentes**

La partición equivalente es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar (Pressman, 2005).

## **Casos de pruebas de caja negra**

Para la realización de las pruebas de caja negra se generaron los artefactos "*Diseño de casos de pruebas*" Por cada requerimiento funcional del sistema se generó un documento en donde se recogen todos los datos necesarios para probar la interfaz. Para ver los diseños de casos de pruebas en su totalidad dirigirse al **Expediente del proyecto Egreso**

## **Resultado de las pruebas de caja negra**

Al aplicar la prueba de caja negra se alcanzaron los siguientes resultados en cada una de las iteraciones, como se muestra en la **Tabla 7**. Las no conformidades encontradas fueron resueltas en su totalidad.



Tabla 7: No conformidades por iteraciones caja negra.

Iteraciones	N° No conformidades	Asociadas a
1ra	19	Errores de interfaz, de validación y ortografía.
2da	9	Errores de interfaz, de validación y ortografía.
3ra	0	-

### 3.6 Conclusiones del capítulo

En este capítulo se detallaron los paradigmas de programación y los estándares de codificación empleados en la implementación de la solución. Se describieron las técnicas para validar los requisitos y las pruebas utilizadas para asegurar la calidad del *software*. Los resultados fueron satisfactorios, asegurando que la solución implementada no contiene errores y tiene la aceptación requerida.



## CONCLUSIONES GENERALES

Una vez finalizada la fundamentación teórica de la investigación, el diseño, implementación y validación de la solución informática propuesta para la gestión del proceso de Ubicación Laboral en la UCI, se arribaron a las siguientes conclusiones:

- El análisis realizado a diferentes sistemas informáticos para la ubicación laboral en el ámbito nacional e internacional permitió fundamentar las bases teóricas de la investigación y demostró que los sistemas estudiados carecen de funcionalidades necesarias para la gestión del proceso de Ubicación Laboral en la UCI. Sin embargo, poseen características que sirvieron como base para elaborar la propuesta de solución.
- Con la aplicación de la metodología desarrollo de *software* DAC se logró guiar el desarrollo del módulo de Ubicación Laboral, documentar la investigación y obtener los artefactos propuestos, quedando registrados en el Expediente del Proyecto Egreso.
- Haciendo uso de las tecnologías establecidas por el Grupo de arquitectura de la DIN se logró la correcta implementación del módulo de Ubicación Laboral y su integración con el subsistema de Egreso del SGU, facilitando la gestión del proceso de Ubicación Laboral en la UCI, la impresión de boletas y la toma efectiva de decisiones.
- Los resultados obtenidos mediante las pruebas realizadas, demostraron el correcto cumplimiento de los requisitos del *software*, la estabilidad y la calidad requerida para la propuesta de solución.



## RECOMENDACIONES

Desarrollar futuras versiones del módulo de Ubicación Laboral del subsistema de Egreso que incorporen nuevas funcionalidades que perfeccionen el trabajo con el mismo, como:

- ❖ Generar automáticamente una propuesta de asignación de las plazas laborales.
- ❖ Obtener los indicadores de integralidad del sistema DataFEU.
- ❖ Generar dinámicamente los reportes de ubicación laboral.

**BIBLIOGRAFÍA REFERENCIADA**

- Alegret, Dr. C. Fernando Vecino. 1995.** Conferencia Mundial de Educación Superior. *Conferencia Mundial de Educación Superior*. 1995.
- Álvarez, Miguel Angel. 2013.** Introducción a HTML. *Introducción a HTML*. 2013.
- . **2010.** Manual de Codelgniter. *Manual de Codelgniter*. 2010.
- Baños, Mara Julia Leandro, Quintana, Leyanis Santiesteban y López, Surayne Torres. 2008.** Biblioteca de la Universidad de las Ciencias Informáticas. *Sistema de Gestión de la Ubicación Laboral de los Egresados de la UCI*. [En línea] 2008. [Citado el: Marzo 08, 2014.] <http://catalogoenlinea.uci.cu/cgi-bin/koha/opac-detail.pl?biblionumber=6732>.
- Bonanata, Ing. Maximiliano. 2000.** *Programación y algoritmos*. Buenos Aires : MP Ediciones S.A, 2000.
- Burbeck, Steve. 2010.** Application programming in Smalltalk-80: How to use Model-View-Controller (MVC). [En línea] 2010. [Citado el: 04 08, 2014.] <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>.
- Comunidad PostgreSQL. 2013.** [pgadmin.org](http://pgadmin.org). *pgadmin.org*. [En línea] 2013. [Citado el: 12 13, 2013.] <http://www.pgadmin.org/features.php>.
- Eguiluz, Javier. 2013.** *Introducción a JavaScript*. 2013.
- EllisLab. 2002-2013.** [En línea] 2002-2013. [Citado el: 12 10, 2013.] [http://ellislab.com/codeigniter/user-guide/overview/at\\_a\\_glance.html](http://ellislab.com/codeigniter/user-guide/overview/at_a_glance.html).
- Espí Lacomba, Nora, y otros. 2009.** La Educación Superior en Cuba 2005-2009. Marzo 2009.
- Félix Alvaro del Castillo, San. 2000.** El servidor de web Apache: Introducción práctica. *El servidor de web Apache: Introducción práctica*. [En línea] 2000. [Citado el: 12 12, 2013.] <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/index.html>.
- Gamma, Erich, y otros. 1994.** *Design Patterns: Elements of Reusable Object Oriented Software*. 1994.
- Garayo, Ing. Jon Gómez García de. 2010.** [Nosolunix.com](http://www.nosolunix.com). *Nosolunix.com*. [En línea] Enero 2010. [Citado el: 04 12, 2014.] <http://www.nosolunix.com/2010/02/herramientas-test-software.html#more>.
- García Vidal, Yanio. 2012.** *Documento de Arquitectura de Gestión Universitaria*. La Habana : s.n., 2012.
- Gestión de la información en las organizaciones. Principios, conceptos y aplicaciones.* **2011.** Santiago de Cuba : Empresa gráfica Haydee Santamaría, Palma Soriano, 2011.
- Empresa gráfica Haydee Santamaría. 2007.** Palma Soriano : Empresa gráfica Haydee Santamaría, 2007.
- Giraldo, O. P. 2007.** *Ingeniería de Requisitos*. 2007. Vol. 13.



- Larman, Craig. 1999.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* México : Editorial Prentice Hall Hispanoamericana, S.A, 1999. 2da Edición.
- Manzur, Sergio. 2013.** Cursos Mexired. *Cursos Mexired.* [En línea] 2013. [Citado el: 12 10, 2013.] <http://www.mexired.com/blog/que-es-jquery/>.
- Martin's, James. 1990.** *Information Engineering.* Prentice Hall : s.n., 1990.
- Méndez, Ing. Alelí Sánchez. 2013.** *Proceso de desarrollo software. Metodología DAC.* La Habana : s.n., 2013.
- Moreno, Juan Manuel Nieto. 2002.** *Introducción a la Programación Orientada a Aspectos.* España : s.n., 2002.
- NetBeans Community. 2013.** netbeans.org. *netbeans.org.* [En línea] 2013. [Citado el: 12 12, 2013.] <https://netbeans.org/index.html>.
- Orfali, Robert and Harkey, Dan. 1998.** *Client/Server Programming with Java and CORBA.* 1998.
- Pressman, Roger. 2001.** *Ingeniería de Software. Un enfoque práctico.* Madrid : s.n., 2001. Quinta Edición.
- . 2005. *Ingeniería del Software. Un enfoque practico.* 2005.
- Sommerville, Ian. 2004.** *Ingeniería de Software.* 2004. 7ma Edición.
- The jQuery Foundation. 2002.** JQuery. *JQuery.* [En línea] 2002. [Citado el: Abril 02, 2014.] <http://jquery.com/>.
- Universidad Externado de Colombia. 2013.** Universidad Externado de Colombia. 2013. [Citado el: Febrero 16, 2014.] <http://portal.uexternado.edu.co/irj/portal/anonymous?NavigationTarget=navurl://3830899b695925fc2973cef6763e68d7>.
- Vázquez, José Antonio Gallego. 2003.** *Desarrollo web con PHP y MySQL.* Madrid : ANAYA Multimedia, 2003.
- Woodman L. 1985.** *Information management in large organizations. Information management from strategies to action.* London : ASLIB, 1985.
- Yuen, P. K. y Lau, V. 2004.** *Practical Web Technologies.* La Habana, Cuba : Félix Varela, 2004.

**BIBLIOGRAFIA CONSULTADA**

- Alegret, Dr. C. Fernando Vecino. 1995.** Conferencia Mundial de Educación Superior. *Conferencia Mundial de Educación Superior*. 1995.
- Álvarez, Miguel Angel. 2013.** Introducción a HTML. *Introducción a HTML*. 2013.
- . **2010.** Manual de Codelgniter. *Manual de Codelgniter*. 2010.
- Álvarez, Saeret Llerena. 2013.** Biblioteca de la Universidad de las Ciencias Informáticas. *Solución informática para la gestión de ubicación laboral en la Universidad de las Ciencias Informáticas*. [En línea] 2013. [Citado el: Marzo 10, 2014.] [http://bibliodoc.uci.cu/RDigitales/2013/septiembre/28/TD\\_06591\\_13.pdf](http://bibliodoc.uci.cu/RDigitales/2013/septiembre/28/TD_06591_13.pdf).
- Baños, Mara Julia Leandro, Quintana, Leyanis Santiesteban y López, Surayne Torres. 2008.** Biblioteca de la Universidad de las Ciencias Informáticas. *Sistema de Gestión de la Ubicación Laboral de los Egresados de la UCI*. [En línea] 2008. [Citado el: Marzo 08, 2014.] <http://catalogoenlinea.uci.cu/cgi-bin/koha/opac-detail.pl?biblionumber=6732>.
- Bonanata, Ing. Maximiliano. 2000.** *Programación y algoritmos*. Buenos Aires : MP Ediciones S.A, 2000.
- Burbeck, Steve. 2010.** Application programming in Smalltalk-80: How to use Model-View-Controller (MVC). [En línea] 2010. [Citado el: 04 08, 2014.] <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>.
- Comunidad PostgreSQL. 2013.** [pgadmin.org](http://pgadmin.org). *pgadmin.org*. [En línea] 2013. [Citado el: 12 13, 2013.] <http://www.pgadmin.org/features.php>.
- Eguiluz, Javier. 2013.** *Introducción a JavaScript*. 2013.
- EllisLab. 2002-2013.** [En línea] 2002-2013. [Citado el: 12 10, 2013.] [http://ellislab.com/codeigniter/user-guide/overview/at\\_a\\_glance.html](http://ellislab.com/codeigniter/user-guide/overview/at_a_glance.html).
- Entorno Virtual de Aprendizaje. EVA. 2012.** Entorno Virtual de Aprendizaje. *Entorno Virtual de Aprendizaje*. [En línea] 2012. [Citado el: 04 09, 2013.] [http://eva.uci.cu/file.php/180/2.\\_Clases/Tema\\_1/Materiales\\_basicos/4.Patrones\\_de\\_diseno\\_de\\_BD.pdf](http://eva.uci.cu/file.php/180/2._Clases/Tema_1/Materiales_basicos/4.Patrones_de_diseno_de_BD.pdf).
- Espí Lacomba, Nora, y otros. 2009.** La Educación Superior en Cuba 2005-2009. Marzo 2009.
- Evolution Solutions CO. 2013.** EVOLUTION SOLUTIONS CO. 2013.
- Félix Alvaro del Castillo, San. 2000.** El servidor de web Apache: Introducción práctica. *El servidor de web Apache: Introducción práctica*. [En línea] 2000. [Citado el: 12 12, 2013.] <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/index.html>.
- Gamma, Erich, y otros. 1994.** *Design Patterns: Elements of Reusable Object Oriented Software*. 1994.



- Garayo, Ing. Jon Gómez García de. 2010.** Nosolounix.com. *Nosolounix.com*. [En línea] Enero 2010. [Citado el: 04 12, 2014.] <http://www.nosolounix.com/2010/02/herramientas-test-software.html#more>.
- García Vidal, Yanio. 2012.** *Documento de Arquitectura de Gestión Universitaria*. La Habana : s.n., 2012.
- Gestión de la información en las organizaciones. Principios, conceptos y aplicaciones. 2011.* Santiago de Cuba : Empresa gráfica Haydee Santamaría, Palma Soriano, 2011.
- Empresa gráfica Haydee Santamaría. 2007.** Palma Soriano : Empresa gráfica Haydee Santamaría, 2007.
- Giraldo, O. P. 2007.** *Ingeniería de Requisitos*. 2007. Vol. 13.
- González, Marihelem Céspedes y Serrano, Lázaro Reyes. 2012.** Biblioteca de la Universidad de las Ciencias Informáticas. *Desarrollo del sistema de gestión de la Dirección de ubicación laboral de la Universidad de las Ciencias Informáticas*. [En línea] Junio 2012. [Citado el: Marzo 10, 2014.] [http://bibliodoc.uci.cu/RDigitales/2013/febrero/20/TD\\_05957\\_12.PDF](http://bibliodoc.uci.cu/RDigitales/2013/febrero/20/TD_05957_12.PDF).
- Larman, Craig. 1999.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México : Editorial Prentice Hall Hispanoamericana, S.A, 1999. 2da Edición.
- Manzur, Sergio. 2013.** Cursos Mexired. *Cursos Mexired*. [En línea] 2013. [Citado el: 12 10, 2013.] <http://www.mexired.com/blog/que-es-jquery/>.
- Martin's, James. 1990.** *Information Engineering*. Prentice Hall : s.n., 1990.
- Méndez, Ing. Alelí Sánchez. 2013.** *Proceso de desarrollo software. Metodología DAC*. La Habana : s.n., 2013.
- Montes, Maylén García. 2012.** Hablando en forma correcta. [En línea] Marzo 26, 2012. [Citado el: Febrero 01, 2014.] <http://hablandoenformacorrecta.blogspot.com/2012/03/las-tics-en-la-educacion-superior.html>.
- Moreno, Juan Manuel Nieto. 2002.** *Introducción a la Programación Orientada a Aspectos*. España : s.n., 2002.
- NetBeans Community. 2013.** netbeans.org. *netbeans.org*. [En línea] 2013. [Citado el: 12 12, 2013.] <https://netbeans.org/index.html>.
- Object Managed Group. 2013.** omg.org. *omg.org*. [En línea] 2013. [Citado el: 12 20, 2013.] [www.omg.org](http://www.omg.org).
- Orfali, Robert and Harkey, Dan. 1998.** *Client/Server Programming with Java and CORBA*. 1998.
- Pressman, Roger. 2001.** *Ingeniería de Software. Un enfoque práctico*. Madrid : s.n., 2001. Quinta Edición.
- . 2005. *Ingeniería del Software. Un enfoque practico*. 2005.



**Sommerville, Ian. 2004.** *Ingeniería de Software*. 2004. 7ma Edición.

*Tendencias internacionales de la Educación Superior. Memorias Universidad. 2006.* 2006.

**The jQuery Foundation. 2002.** JQuery. *JQuery*. [En línea] 2002. [Citado el: Abril 02, 2014.] <http://jquery.com/>.

**The PHP Group. 2001-2013.** php.net. *php.net*. [En línea] 2001-2013. [Citado el: 12 20, 2013.] [www.php.net](http://www.php.net).

**The PostgreSQL Global Development Group. 2010.** PostgreSQL. *PostgreSQL*. [En línea] 2010. [Citado el: 12 10, 2013.] <http://www.postgresql.org/about/>.

**Universidad de Externado de Colombia. 2013.** Universidad de Externado de Colombia. 2013. [Citado el: Febrero 16, 2014.] <http://portal.uexternado.edu.co/irj/portal/anonymous?NavigationTarget=navurl://3830899b695925fc2973cef6763e68d7>.

**Vázquez, José Antonio Gallego. 2003.** *Desarrollo web con PHP y MySQL*. Madrid : ANAYA Multimedia, 2003.

**Visual Paradigm Company. 2013.** UML, BPMN and Enterprise Architecture Tool for Software Development. *UML, BPMN and Enterprise Architecture Tool for Software Development*. [En línea] 02 21, 2013. [Citado el: 04 25, 2014.] <http://www.visual-paradigm.com>.

**Woodman L. 1985.** *Information management in large organizations. Information management from strategies to action*. London : ASLIB, 1985.

**Yuen, P. K. y Lau, V. 2004.** *Practical Web Technologies*. La Habana, Cuba : Félix Varela, 2004.



## GLOSARIO DE TÉRMINOS

**Artefacto:** es la especificación de un componente físico de información que es usado o producido por un proceso de desarrollo de software, o por el desarrollo y operación de un sistema.

**Demanda:** En el contexto del proceso de ubicación laboral, una demanda no es más que la cantidad de plazas que solicita una empresa al Ministerio de Trabajo y Seguridad Social del país.

**Egreso:** todo aquello que egresa o sale de un lugar o espacio determinado. Es la finalización de una carrera o proyecto ya que es allí cuando se considera que una persona egresa o deja una institución educativa. Egresar es justamente uno de los momentos más importantes en la vida de un estudiante debido a que es la finalización de todo el esfuerzo.

**Flujo:** secuencia de los aspectos operacionales de una actividad de trabajo: como se estructuran las tareas, como se realizan y cuál es su orden de ejecución.

**IDE:** El Entorno de Desarrollo Integrado es un programa compuesto por un conjunto de herramientas para un programador.

**Identación:** es un anglicismo (de la palabra inglesa *indentation*) de uso común en informática que significa mover un bloque de texto hacia la derecha insertando espacios o tabuladores para separarlo del texto adyacente, lo que en el ámbito de la imprenta se ha denominado siempre como sangrado o sangría.

**Información:** es un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.

**Organismo:** es una unidad que se encarga de una determinada función hacia la sociedad. Es una entidad social con capacidad jurídica para realizar fines específicos, estables y estructurados formalmente, de tal manera que permita la eficiencia del trabajo grupal en la consecución de objetivos dentro de una misión social preestablecida.

**Proceso:** conjunto de actividades relacionadas lógicamente que producen una salida o resultado.

**JQuery UI:** es una biblioteca de componentes para el *framework* jQuery que le añaden un conjunto de *plugins*, *widgets* y efectos visuales para la creación de aplicaciones web.

**Ubicación laboral:** Proceso mediante el cual se asigna una plaza laboral a cada estudiante egresado de las universidades cubanas.

# ANEXOS

## Anexo 1: Diagrama de proceso de negocio

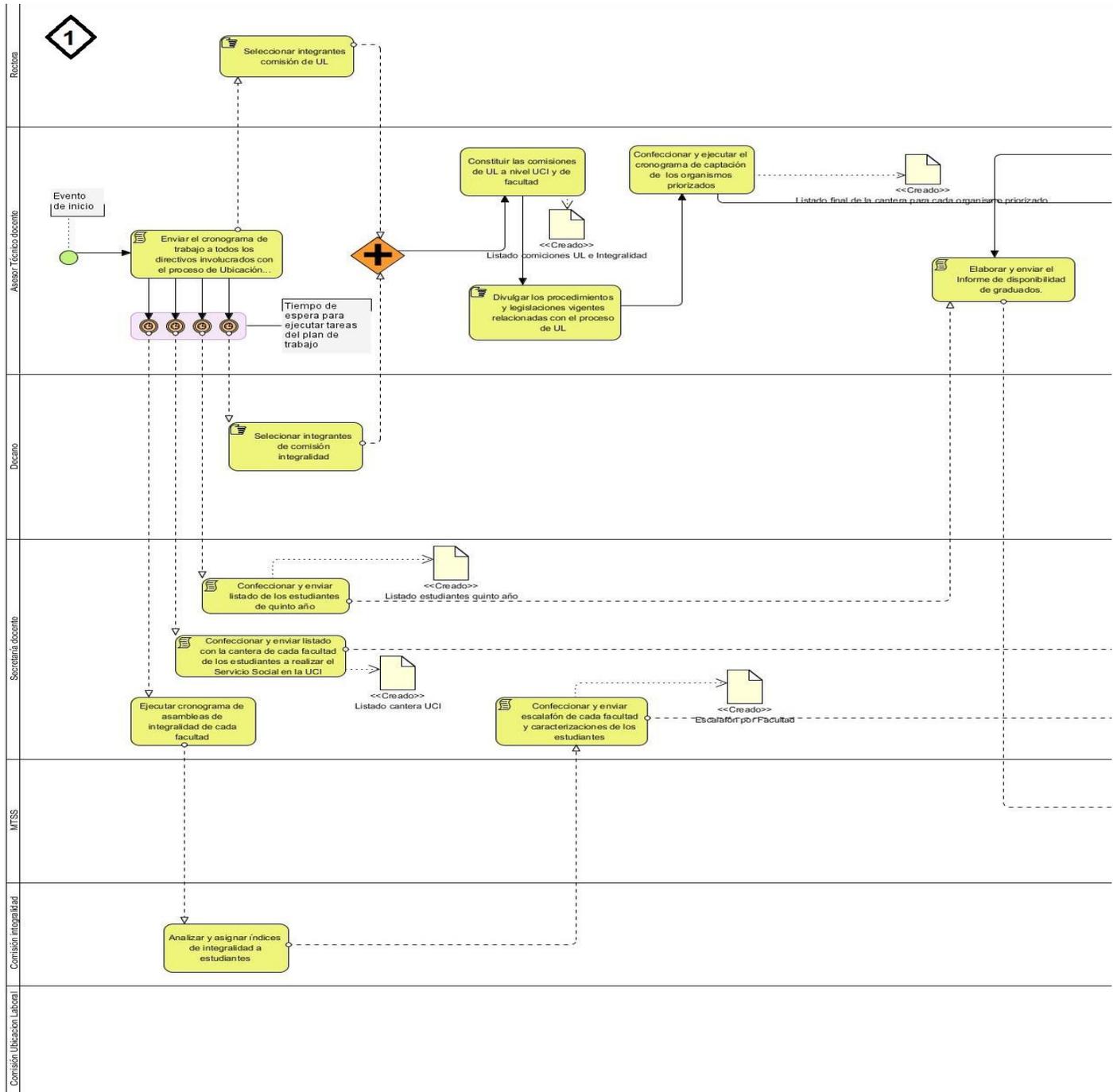


Figura 18: Proceso de negocio de ubicación laboral parte 1.

2

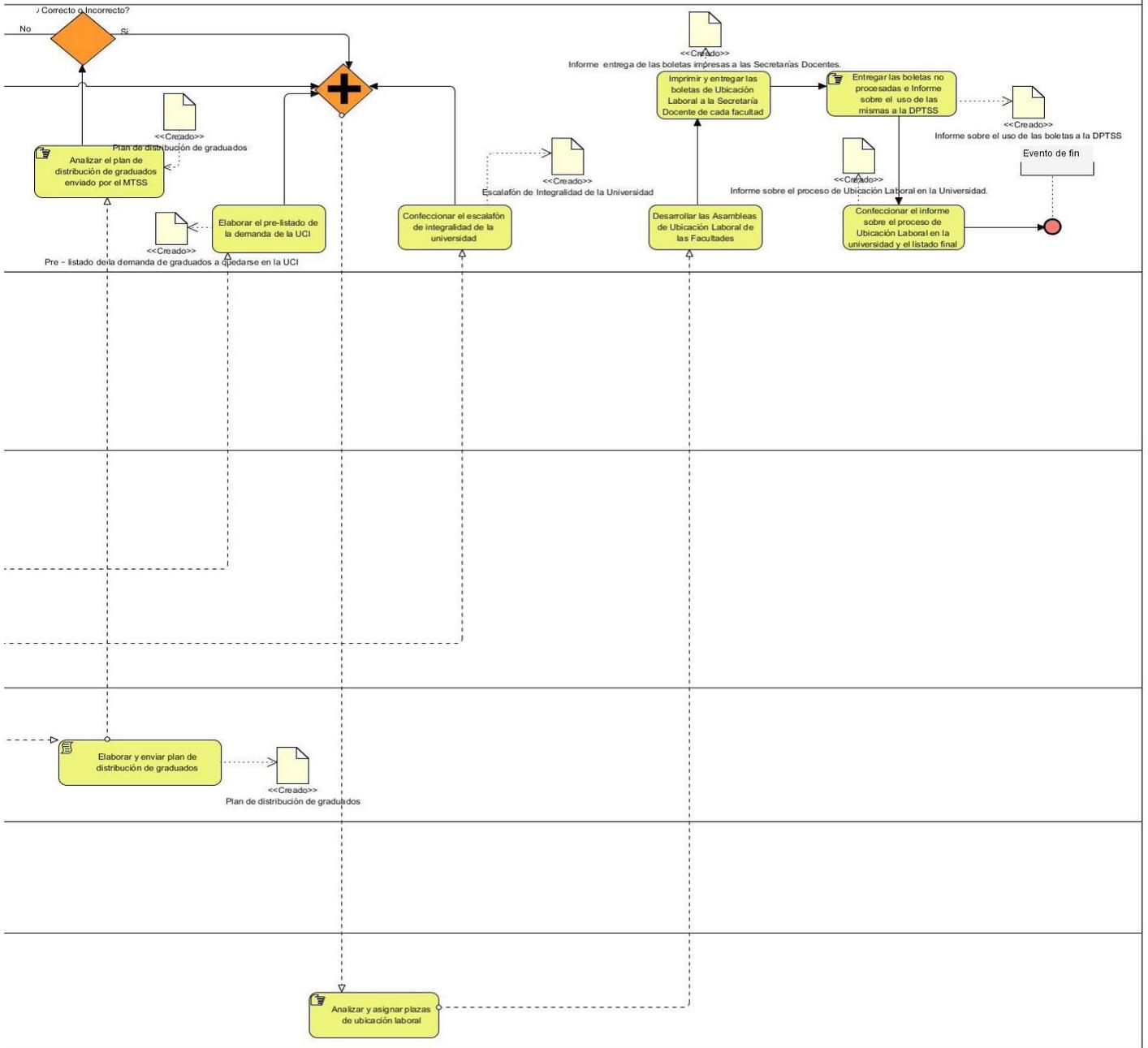


Figura 19: Proceso de negocio de ubicación laboral parte 2.

## **Anexo 2: Reglas del negocio**

A continuación se enumeran las reglas del negocio del proceso de Ubicación Laboral en la UCI:

1. La Comisión se identifica con el nombre de la universidad o la facultad, en dependencia de la tipología que tenga la misma.
2. Los decanos y el rector son los únicos responsables de las comisiones que les corresponde.
3. El Asesor Técnico Docente de la Oficina de Ubicación Laboral es el responsable de crear las comisiones e informar a sus presidentes que deben asignarle a cada integrante de la comisión un rol específico.

### **La Comisión de Ubicación Laboral de la Universidad está integrada por:**

- Presidente: Rector.
- Secretario Ejecutivo: Vice-rector.
- Especialista de ubicación laboral: Asesor Técnico Docente de Ubicación Laboral.
- Directivo de la Federación Estudiantil Universitaria de la universidad: Presidente o directivo designado por el presidente.

### **La Comisión de integralidad de la Facultad está integrada por:**

- Presidente: Decano.
  - Secretario Ejecutivo: Vice-decano.
  - Coordinador de 5º año: Profesor.
  - Profesores guías del año: Profesor.
  - Directivo de la Federación Estudiantil Universitaria de la facultad: Presidente o directivo designado por el presidente.
  - Invitados
4. No se repite ningún rol en una comisión, excepto Profesores Guías del año.
  5. El proceso de Ubicación Laboral solo se le realiza a los estudiantes civiles graduados del curso regular diurno.
  6. Se le asigna una plaza laboral a cada estudiante.
  7. La plaza que se le asigna al estudiante es válida hasta el mes de diciembre del mismo año fiscal.
  8. El código de la empresa está compuesto por ocho caracteres, siendo los tres primeros el código del organismo al que pertenece.

9. El registro de demanda se confecciona para cada empresa, permitiendo almacenar la demanda enviada por el MTSS.
10. El Asesor Técnico Docente de la Oficina de Ubicación Laboral es el único responsable sobre el registro de demandas.
11. El decano de la facultad solo podrá ver la comisión de su misma facultad.
12. Los estudiantes del MININT o de la FAR no se tienen en cuenta en el proceso de asignación de plaza.
13. Para asignarle plaza a un estudiante, este debe haber terminado las asignaturas de su plan de estudio, aunque se puede asignar plaza a un estudiante que haya suspendido el ejercicio de culminación de estudio (tesis).
14. Se diferenciarán las empresas priorizadas de las no priorizadas.
15. Se realizará la captación de los estudiantes para las entidades priorizadas antes de realizar la asignación de las plazas laborales de las entidades no priorizadas.

### Anexo 3: Descripción de requisitos

Tabla 8: Descripción de requisito. Mostrar comisión de ubicación laboral.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL1	Mostrar comisión de ubicación laboral.	<p>El requisito permite mostrar un listado de las comisiones registradas en el sistema, además muestra las opciones: Modificar, Asignar rol a una persona miembro de una comisión y Ver detalles en el área de íconos internos, así como la opción Crear del área de íconos flotantes.</p> <p>Permite realizar una búsqueda de las comisiones registrados a partir del nombre.</p> <p>El requisito comienza cuando el administrador, el Asesor Técnico Docente, los decanos o la rectora selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Comisiones</b> del menú de funcionalidades <b>Diseño de Ubicación Laboral</b>.</p>	Media	Alta
<b>Prototipo</b>				

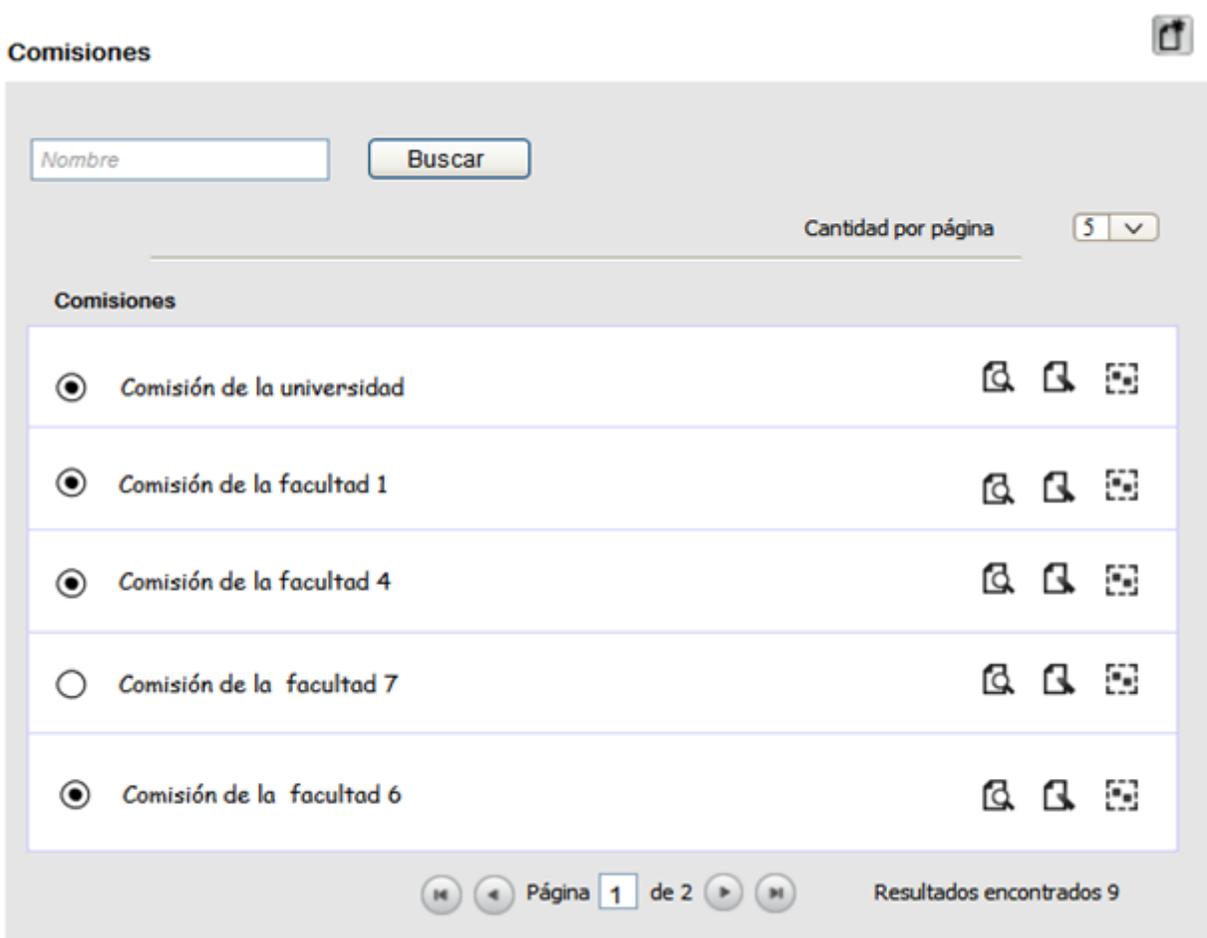
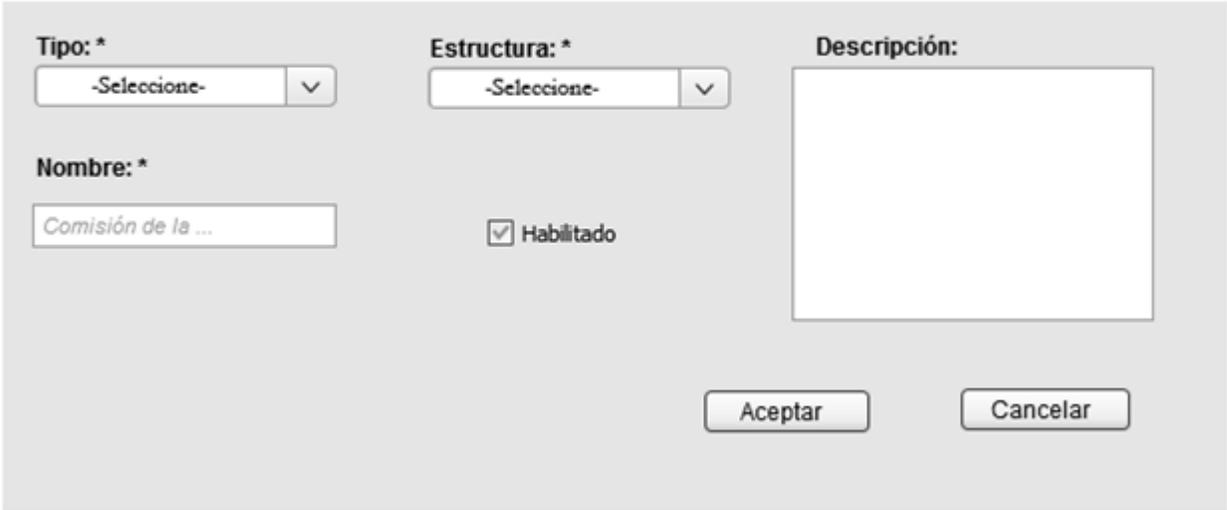
	 <p><b>Comisiones</b></p> <p>Nombre <input type="text"/> <input type="button" value="Buscar"/></p> <p>Cantidad por página <input type="text" value="5"/> ▼</p> <p><b>Comisiones</b></p> <ul style="list-style-type: none"> <li><input checked="" type="radio"/> Comisión de la universidad</li> <li><input checked="" type="radio"/> Comisión de la facultad 1</li> <li><input checked="" type="radio"/> Comisión de la facultad 4</li> <li><input type="radio"/> Comisión de la facultad 7</li> <li><input checked="" type="radio"/> Comisión de la facultad 6</li> </ul> <p>Página 1 de 2 Resultados encontrados 9</p>
	<p align="center"><b>Prototipo Mostrar comisión de ubicación laboral.</b></p> <p><b>Observaciones</b></p> <ol style="list-style-type: none"> <li>1. En caso de que no haya registrado en el sistema comisiones, se muestra el listado vacío.</li> <li>2. Se debe indicar si está habilitada la comisión.</li> <li>3. Interactúa con esta acción el administrador del sistema, el Asesor Técnico Docente, los decanos y la rectora.</li> </ol>

Tabla 9: Descripción de requisito. Crear comisión de ubicación laboral

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL2	Crear comisión de ubicación laboral.	El requisito permite crear comisiones de ubicación laboral. Se muestran los campos a llenar: nombre, tipo, estructura,	Media	Alta

		<p>descripción y el campo de selección habilitado, además se muestra en el área de íconos flotantes la opción Listar. Una vez creada la comisión se actualiza el listado y se muestra un mensaje de información “El elemento ha sido creado satisfactoriamente”.</p> <p>El requisito comienza cuando el administrador, el Asesor Técnico Docente, los decanos o la rectora selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Comisiones</b> del menú de funcionalidades <b>Diseño de Ubicación Laboral</b> y selecciona en el área de íconos flotantes la opción <b>Crear comisión</b>.</p>		
<b>Prototipo</b>				
 <p style="text-align: center;"><b>Prototipo Crear comisión de ubicación laboral.</b></p>				
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	Tipo	Character varying	Campo obligatorio. El campo permite seleccionar el	

			tipo de comisión que se desea crear, universidad o facultad.
	Estructura	Character varying	Campo obligatorio. Permite seleccionar la estructura a la que pertenece la comisión creada.
	Nombre	Character varying	Campo obligatorio. Admite a partir de 2 caracteres excepto % y comillas simples. La cantidad de caracteres permitidos para una palabra es de 30. La cantidad total de caracteres permitidos es 250.

	Estado(Habilitado/ Deshabilitado)	Boolean	Campo de selección que permite especificar si una comisión está habilitada o deshabilitada.
	Descripción	Text	Admite a partir de 0 caracteres excepto % y comillas simples. La cantidad de caracteres permitidos para una palabra es de 30. La cantidad total de caracteres permitidos es 1000.
	<b>Observaciones</b>	<ol style="list-style-type: none"> <li>1. En caso que el elemento ya exista se muestra un mensaje de error "El elemento ya existe".</li> <li>2. Si se crea satisfactoriamente el elemento se mostrará un mensaje de información "El elemento ha sido creado satisfactoriamente".</li> <li>3. En caso de cancelar la acción se muestra un mensaje de confirmación "¿Está seguro de cancelar la acción?". En caso que el usuario presione el botón Aceptar se regresa al listado de comisiones. Si el usuario presiona el botón Cancelar se mantiene en la misma vista.</li> <li>4. En caso de que introduzca datos incorrectos (porcentaje % y/o comillas simples ')</li> </ol>	

		<p>se muestra un mensaje de error “No se admiten apóstrofes ni porcientos”.</p> <p>5. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje de error de color rojo “Campo requerido” en el campo que debe ser llenado obligatoriamente.</p> <p>6. Si se introduce más caracteres de los permitidos para una palabra el sistema muestra el mensaje “Ha excedido el número de letras permitidas para una palabra”.</p> <p>7. Si se trata de introducir más de los caracteres permitidos para los campos el sistema no permite seguir introduciéndolos.</p> <p>8. Intervienen en esta acción el administrador del sistema, los decanos, el Asesor Técnico Docente y la rectora.</p>
--	--	---

Tabla 10: Descripción de requisito. Modificar comisión de ubicación laboral.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL3	Modificar comisión de ubicación laboral.	<p>El requisito permite modificar comisiones de ubicación laboral. Se muestran los campos a llenar: nombre, tipo, estructura, descripción y el campo de selección habilitado, además se muestra en el área de íconos flotantes la opción Listar. Una vez modificada la comisión se actualiza el listado y se muestra un mensaje de información “El elemento ha sido modificado satisfactoriamente”.</p> <p>El requisito comienza cuando el administrador, el Asesor Técnico Docente, los decanos o la rectora selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Comisiones</b> del menú de funcionalidades <b>Diseño de Ubicación Laboral</b> y selecciona en el área de íconos internos la opción <b>Modificar comisión</b></p>	Media	Media
	<b>Prototipo</b>			

### Modificar Comisión

**Prototipo Modificar comisión de ubicación laboral.**

Campos	Tipos de Datos	Reglas o Restricciones
Tipo	Character varying	Campo obligatorio. El campo permite seleccionar el tipo de comisión que se desea crear, universidad o facultad.
Estructura	Character varying	Campo obligatorio. Permite seleccionar la estructura a la que pertenece la comisión creada.
Nombre	Character varying	Campo obligatorio. Admite a partir de 2 caracteres excepto % y comillas simples. La cantidad de caracteres permitidos para una palabra es de 30. La cantidad total de caracteres permitidos es 250.

	Estado(Habilitado/ Deshabilitado)	Boolean	Campo de selección que permite especificar si una comisión está habilitada o deshabilitada.
	Descripción	Text	Admite a partir de 0 caracteres excepto % y comillas simples. La cantidad de caracteres permitidos para una palabra es de 30. La cantidad total de caracteres permitidos es 1000.
	<b>Observaciones</b>	<ol style="list-style-type: none"> <li>1. En caso que el elemento ya exista se muestra un mensaje de error “El elemento ya existe”.</li> <li>2. Si se modifica satisfactoriamente el elemento se mostrará un mensaje de información “El elemento ha sido modificado satisfactoriamente”.</li> <li>3. En caso de cancelar la acción se muestra un mensaje de confirmación “¿Está seguro de cancelar la acción?”. En caso que el usuario presione el botón Aceptar se regresa al listado de comisiones. Si el usuario presiona el botón Cancelar se mantiene en la misma vista.</li> <li>4. En caso de que introduzca datos incorrectos (porcentaje % y/o comillas simples ` `) se muestra un mensaje de error “No se admiten apóstrofes ni porcentajes”.</li> <li>5. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje de error de color rojo “Campo requerido” en el campo que debe ser llenado obligatorio.</li> <li>6. Si se introduce más caracteres de los permitidos para una palabra el sistema muestra el mensaje “Ha excedido el número de letras permitidas para una palabra”.</li> <li>7. Si se trata de introducir más de los caracteres permitidos para los campos el sistema no permite seguir introduciéndolos.</li> <li>8. Intervienen en esta acción el administrador del sistema, el Asesor Técnico Docente, los decanos y la rectora.</li> </ol>	

Tabla 11: Descripción de requisito. Ver detalles comisión de ubicación laboral.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL4	Ver detalles comisión de ubicación laboral.	<p>El requisito permite ver detalles de la comisión de ubicación laboral seleccionada por el usuario. Se muestran los datos en forma de ventana emergente.</p> <p>El requisito comienza cuando el administrador, el Asesor Técnico Docente, los decanos o el rector selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Comisiones</b> del menú de funcionalidades <b>Diseño de Ubicación Laboral</b> y selecciona en el área de íconos internos la opción <b>Ver detalles</b> de la comisión deseada.</p>	Baja	Baja
<b>Prototipo</b>				

	<div style="border: 1px solid gray; padding: 10px;"> <div style="text-align: right; border-bottom: 1px solid gray; padding-bottom: 5px;"><b>Detalles de Comisión</b> <span style="float: right;">✕</span></div> <p><b>Tipo:</b> Facultad  <b>Estructura:</b> Facultad 1  <b>Nombre:</b> Comisión de la Facultad 1  <b>Estado:</b> Habilitado  <b>Descripción:</b></p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <p style="text-align: right;"><b>Miembros de la Comisión de la Facultad 1</b> Cantidad por página <span style="border: 1px solid gray; padding: 2px;">3</span> ▾</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 10%;">Nombre y Apellidos</th> <th style="width: 45%;">Cargo</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><input checked="" type="radio"/></td> <td> <span style="margin-left: 5px;">Damián Cervantes Rondon</span></td> <td>Decano</td> </tr> <tr> <td style="text-align: center;"><input checked="" type="radio"/></td> <td> <span style="margin-left: 5px;">Yenisleidi Carriaga Cristo</span></td> <td>Vicedecano</td> </tr> <tr> <td style="text-align: center;"><input checked="" type="radio"/></td> <td> <span style="margin-left: 5px;">Nestor A. Delgado Pacheco</span></td> <td>Presidente de la FEU</td> </tr> </tbody> </table> </div> <div style="text-align: center; margin-top: 10px;"><input type="button" value="Cerrar"/></div> </div> <p style="text-align: center;"><b>Prototipo Ver detalles de comisión de ubicación laboral.</b></p>		Nombre y Apellidos	Cargo	<input checked="" type="radio"/>	<span style="margin-left: 5px;">Damián Cervantes Rondon</span>	Decano	<input checked="" type="radio"/>	<span style="margin-left: 5px;">Yenisleidi Carriaga Cristo</span>	Vicedecano	<input checked="" type="radio"/>	<span style="margin-left: 5px;">Nestor A. Delgado Pacheco</span>	Presidente de la FEU
	Nombre y Apellidos	Cargo											
<input checked="" type="radio"/>	<span style="margin-left: 5px;">Damián Cervantes Rondon</span>	Decano											
<input checked="" type="radio"/>	<span style="margin-left: 5px;">Yenisleidi Carriaga Cristo</span>	Vicedecano											
<input checked="" type="radio"/>	<span style="margin-left: 5px;">Nestor A. Delgado Pacheco</span>	Presidente de la FEU											
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;"><b>Observaciones</b></td> <td>1. Intervienen en esta acción el administrador del sistema, el Asesor Técnico Docente, los decanos y la rectora.</td> </tr> </table>	<b>Observaciones</b>	1. Intervienen en esta acción el administrador del sistema, el Asesor Técnico Docente, los decanos y la rectora.										
<b>Observaciones</b>	1. Intervienen en esta acción el administrador del sistema, el Asesor Técnico Docente, los decanos y la rectora.												

Tabla 12: Descripción de requisito. Asociar persona a comisión.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL5	Asociar persona a comisión.	El requisito permite asociar personas en las comisiones Se selecciona la opción <b>asociar persona a comisión</b> en la barra de íconos internos a partir del listado de comisiones. Una vez seleccionada la opción asociar personas, se muestra un buscador de personas, un campo de selección para	Alta	Alta

		<p>especificar el rol, y la opción de filtrar la búsqueda por cargo, además se muestra en el área de íconos flotantes la opción Listar. Luego se muestra el listado con las personas que coincidan con los parámetros de la búsqueda. Se selecciona la persona a la que se le asignará el rol especificado, se selecciona el botón “Asociar” y se muestra el miembro seleccionado en el listado de miembros asociados. Se presiona el botón “Aceptar”, y se guardan los datos.</p> <p>En el listado de los grupos asociados se muestra la opción <b>Eliminar persona de comisión</b>. Además, luego de guardados los datos se muestra el mensaje de información “Los datos han sido asociados satisfactoriamente.”</p>		
<b>Prototipo</b>				

**Desagregar rol a persona de la comisión de la facultad 1**

nombre, apellido, CI, solapín, usuario  Filtrar búsqueda Cargos

Cargo Decano

Cantidad por página 3

	Nombre y Apellidos	Cargo	Área
<input checked="" type="radio"/>	 Damián Cervantes Rondon	Decano	Facultad 1
<input type="radio"/>	 Lidia Ruiz Ortiz	Decano	Facultad 4
<input type="radio"/>	 Mayra Duran Benejam	Decano	Facultad 5

Página 1 de 2 Resultados encontrados 6

**Miembros de la Comisión de la Facultad 1**

	Nombre y Apellidos	Cargo	
<input checked="" type="radio"/>	 Damián Cervantes Rondon	Decano	<input type="button" value="X"/>
<input type="radio"/>	 Yenisleidi Carriaga Cristo	Vicedecano	<input type="button" value="X"/>
<input type="radio"/>	 Nestor A. Delgado Pacheco	Presidente de la FEU	<input type="button" value="X"/>

**Prototipo Asociar persona a comisión.**

Campos	Tipos de Datos	Reglas o Restricciones
Rol	Character varying	Campo obligatorio. Permite seleccionar el rol con el cual se asociará la persona a la comisión.
<b>Observaciones</b>	<ol style="list-style-type: none"> <li>1. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de cancelar la acción?”.</li> <li>2. En caso de que no existan personas que cumplan con los parámetros de la búsqueda, se muestra el listado vacío de igual manera si no existe hasta el momento alguna persona asociada.</li> <li>3. Solo se asociará la persona cuando se Acepta de manera completa la asociación.</li> </ol>	

		<p>4. Cuando este aceptada la asociación es que permitirá desasociar una persona.</p> <p>5. Interactúa con esta acción el administrador del sistema, el Asesor Técnico Docente, los decanos y la rectora.</p>
--	--	---

Tabla 13: Descripción de requisito. Eliminar persona de comisión.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL6	Eliminar persona de comisión.	<p>El requisito permite desasociar personas de las comisiones a las que están asociadas. Para eliminar personas de las comisiones se selecciona la opción Eliminar en la barra de íconos internos del listado de personas asociadas a la comisión. Una vez eliminada la persona se actualiza el listado de personas asociadas a la comisión. Además se muestra en el área de íconos flotantes la opción Listar.</p> <p>El requisito comienza cuando el administrador, el Asesor Técnico Docente, los decanos o la rectora selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Comisiones</b> del menú de funcionalidades <b>Diseño de Ubicación Laboral</b>.</p>	Media	Alta
<b>Prototipo</b>				

	<p style="text-align: center;"><b>Prototipo Eliminar persona de comisión.</b></p>
<p><b>Observaciones</b></p>	<ol style="list-style-type: none"> <li>1. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de cancelar la acción?”.</li> <li>2. Interactúa con esta acción el administrador del sistema, el Asesor Técnico Docente, los decanos y la rectora.</li> <li>3. Cuando se selecciona desasociar inicialmente se elimina del listado de los datos asociados, pero se desasocia cuando se selecciona el botón “Aceptar”.</li> </ol>

Tabla 14: Descripción de requisito. Mostrar entidad.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
----	--------	-------------	-------------	------------------------

RFUL7	Mostrar entidad	<p>El requisito permite mostrar un listado de las entidades registradas en el sistema, además muestra las opciones: Modificar entidad, Ver detalles y Asociar registro de demanda en el área de íconos internos, así como la opción Crear entidad en el área de íconos flotantes.</p> <p>Permite realizar una búsqueda avanzada permitiendo filtrar los resultados por Municipio y Provincia.</p> <p>El requisito comienza cuando el administrador, el Asesor Técnico Docente o la rectora selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Comisiones</b> del menú de funcionalidades <b>Diseño de Ubicación Laboral</b>.</p>	Media	Alta
<b>Prototipo</b>				

	<div style="text-align: right; margin-bottom: 5px;"> </div> <div style="text-align: center;"> <h3>Entidades</h3> </div> <div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 40%;"> <input type="text" value="Nombre"/> </div> <div style="width: 15%; text-align: center;"> <input type="button" value="Buscar"/> </div> <div style="width: 40%; text-align: right;"> <div style="margin-bottom: 5px;">Filtrar búsqueda</div> <input type="text" value="-Seleccione-"/> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="width: 45%;"> <div style="margin-bottom: 5px;">Municipio</div> <input type="text" value="-Seleccione-"/> </div> <div style="width: 45%;"> <div style="margin-bottom: 5px;">Provincia</div> <input type="text" value="-Seleccione-"/> </div> </div> <div style="text-align: right; margin-top: 10px;">       Cantidad por página <input type="text" value="5"/> </div> </div> <div style="margin-top: 10px;"> <h4>Entidades</h4> <table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td style="text-align: center;"><input type="radio"/></td> <td>"DISAIC" Casa Consultora</td> <td style="text-align: right;"> </td> </tr> <tr> <td style="text-align: center;"><input type="radio"/></td> <td>Empresa de Computación</td> <td style="text-align: right;"> </td> </tr> <tr> <td style="text-align: center;"><input type="radio"/></td> <td>Especialista de Ingeniería Construcción y Mantenimiento (EMCODI)</td> <td style="text-align: right;"> </td> </tr> <tr> <td style="text-align: center;"><input type="radio"/></td> <td>Empresa Azucarera Manuel Fajardo</td> <td style="text-align: right;"> </td> </tr> <tr> <td style="text-align: center;"><input type="radio"/></td> <td>ZETI</td> <td style="text-align: right;"> </td> </tr> </tbody> </table> </div> <div style="text-align: center; margin-top: 10px;"> <input type="button" value="«"/> <input type="button" value="◀"/>       Página <input type="text" value="1"/> de 2       <input type="button" value="▶"/> <input type="button" value="»"/> <span style="float: right;">Resultados encontrados 8</span> </div>		<input type="radio"/>	"DISAIC" Casa Consultora		<input type="radio"/>	Empresa de Computación		<input type="radio"/>	Especialista de Ingeniería Construcción y Mantenimiento (EMCODI)		<input type="radio"/>	Empresa Azucarera Manuel Fajardo		<input type="radio"/>	ZETI	
<input type="radio"/>	"DISAIC" Casa Consultora																
<input type="radio"/>	Empresa de Computación																
<input type="radio"/>	Especialista de Ingeniería Construcción y Mantenimiento (EMCODI)																
<input type="radio"/>	Empresa Azucarera Manuel Fajardo																
<input type="radio"/>	ZETI																
	<b>Observaciones</b>	<ul style="list-style-type: none"> <li>En caso de que no haya registrados en el sistema entidades muestra el listado vacío.</li> <li>Se debe indicar si está habilitado la entidad.</li> <li>Interactúa con esta acción el administrador del sistema, el Asesor Técnico Docente y la rectora.</li> </ul>															

**Prototipo Mostrar entidad.**

Tabla 15: Descripción de requisito. Crear entidad.

Nº	Nombre	Descripción	Complejidad	Prioridad
----	--------	-------------	-------------	-----------

				<b>para cliente</b>
RFUL8	Crear entidad.	El requisito permite crear entidades. Se muestran los campos a llenar organismo al que pertenece, código organismo, código entidad, nombre, provincia, municipio, teléfono, dirección de la entidad, descripción y el campo de selección habilitado, además se muestra en el área de íconos flotantes la opción Listar. Una vez creada la entidad se actualiza el listado y se muestra un mensaje de información "El elemento ha sido creado satisfactoriamente".	Media	Alta
<b>Prototipo</b>				
<p><b>Crear entidad</b></p> <p>Organismo al que pertenece: *    Código organismo: *    Descripción:</p> <p>-Seleccione-    Código    [Text Area]</p> <p>Código entidad: *    Nombre: *    [Text]</p> <p>Código    Nombre</p> <p>Provincia: *    Municipio: *    -Seleccione-    [Text]</p> <p>-Seleccione-    [Text]</p> <p>Dirección entidad: *    Teléfono: *    [Text]</p> <p>Dirección    Teléfono</p> <p><input checked="" type="checkbox"/> Habilitado    [Aceptar]    [Cancelar]</p>				
<b>Prototipo Crear entidad.</b>				
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	Organismo al que pertenece	Character varying	Campo obligatorio. Permite seleccionar el nombre del	

			organismo al que pertenece la entidad.
	Código organismo	Integer	Campo obligatorio. Se muestra el campo inactivo.
	Código entidad	Integer	Campo obligatorio. La cantidad de caracteres permitidos para un número es de 10. Los tres primeros caracteres del campo están conformados con el código del organismo al que pertenece.
	Nombre	Character varying	Campo obligatorio. Admite a partir de 2 caracteres excepto % y comillas simples. La cantidad de caracteres permitidos para una palabra es de 30. La cantidad de caracteres permitidos es 250.
	Provincia	Character varying	Campo obligatorio. Permite seleccionar la provincia a la que pertenece la entidad.
	Municipio	Character varying	Campo obligatorio. Permite seleccionar el municipio al que pertenece la entidad. Este campo se mantiene oculto hasta que no se seleccione la provincia.
	Teléfono	Integer	La cantidad de caracteres permitidos para un número es de 10.
	Dirección de la entidad	Character varying	Admite a partir de 2 caracteres excepto % y comillas simples. La cantidad

			de caracteres permitidos para una palabra es de 30. La cantidad de caracteres permitidos es 250.
	Estado(Habilidad o/Deshabilitado)	Boolean	Campo de selección que permite especificar si un rol está habilitado o deshabilitado.
	Descripción	Text	Admite a partir de 0 caracteres excepto % y comillas simples. La cantidad de caracteres permitidos para una palabra es de 30. La cantidad de caracteres permitidos es 1000.
	<b>Observaciones</b>	<ol style="list-style-type: none"> <li>1. En caso que el elemento ya exista se muestra un mensaje de error "El elemento ya existe".</li> <li>2. Si se crea satisfactoriamente el elemento se mostrará un mensaje de información "El elemento ha sido creado satisfactoriamente".</li> <li>3. En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de cancelar la acción?". En caso que el usuario presione el botón Aceptar se regresa al listado de empresas. Si el usuario presiona el botón Cancelar se mantiene en la misma vista.</li> <li>4. En caso de que introduzca datos incorrectos (porcentaje % y comillas simples ´) se muestra un mensaje de error "No se admiten apóstrofes ni porcentajes".</li> <li>5. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje de error de color rojo "Campo requerido" en el campo que debe ser llenado obligatorio.</li> <li>6. Si se introduce más caracteres de los permitidos para una palabra el sistema muestra el mensaje "Ha excedido el número de letras permitidas para una palabra".</li> <li>7. En caso que se introduzca algún carácter que no sea número en el campo código de entidad o de organismo, el sistema muestra el mensaje de error: "Entre solo</li> </ol>	

		<p>números”.</p> <p>8. En dependencia del organismo que se selecciona se muestra el código del organismo.</p> <p>9. Interactúa con esta acción el administrador del sistema, el Asesor Técnico Docente y la rectora.</p>
--	--	--

Tabla 16: Descripción de requisito. Modificar entidad.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL9	Modificar entidad.	El requisito permite modificar las entidades existentes. Se muestran los campos a modificar organismo al que pertenece, código organismo, código entidad, nombre, provincia, municipio, teléfono, dirección de la entidad, descripción y el campo de selección habilitado, además se muestra en el área de íconos flotantes la opción Listar. Una vez modificada la entidad se actualiza el listado y se muestra un mensaje de información “El elemento ha sido modificado satisfactoriamente”.	Media	Alta
	<b>Prototipo</b>			

## Modificar entidad

Organismo al que pertenece: \*

Código organismo: \*

Descripción:

Código entidad: \*

Nombre: \*

Provincia: \*

Municipio: \*

Dirección entidad: \*

Teléfono: \*

Habilitado

Aceptar

Cancelar

### Prototipo Modificar entidad.

Campos	Tipos de Datos	Reglas o Restricciones
Organismo al que pertenece	Character varying	Campo obligatorio. Permite seleccionar el nombre del organismo al que pertenece la entidad.
Código organismo	Integer	Campo obligatorio. Se muestra el campo inactivo.
Código entidad	Integer	Campo obligatorio. La cantidad de caracteres permitidos para un número es de 10. Los tres primeros caracteres del campo están conformados con el código del organismo al que pertenece.

	Nombre	Character varying	Campo obligatorio. Admite a partir de 2 caracteres excepto % y comillas simples. La cantidad de caracteres permitidos para una palabra es de 30. La cantidad de caracteres permitidos es 250.
	Provincia	Character varying	Campo obligatorio. Permite seleccionar la provincia a la que pertenece la entidad.
	Municipio	Character varying	Campo obligatorio. Permite seleccionar el municipio al que pertenece la entidad. Este campo se mantiene oculto hasta que no se seleccione la provincia.
	Teléfono	Integer	La cantidad de caracteres permitidos para un número es de 10.
	Dirección de la empresa	Character varying	Admite a partir de 2 caracteres excepto % y comillas simples. La cantidad de caracteres permitidos para una palabra es de 30. La cantidad de caracteres permitidos es 250.
	Estado(Habilitado/Deshabilitado)	Boolean	Campo de selección que permite especificar si un rol está habilitado o deshabilitado.
	Descripción	Text	Admite a partir de 0 caracteres excepto % y comillas simples. La cantidad

			de caracteres permitidos para una palabra es de 30. La cantidad de caracteres permitidos es 1000.
	<b>Observaciones</b>	<ol style="list-style-type: none"> <li>1. En caso que el elemento ya exista se muestra un mensaje de error “El elemento ya existe”.</li> <li>2. Si se modifica satisfactoriamente el elemento se mostrará un mensaje de información “El elemento ha sido modificado satisfactoriamente”.</li> <li>3. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de cancelar la acción?”. En caso que el usuario presione el botón Aceptar se regresa al listado de empresas. Si el usuario presiona el botón Cancelar se mantiene en la misma vista.</li> <li>4. En caso de que introduzca datos incorrectos (porcentaje % y comillas simples `) se muestra un mensaje de error “No se admiten apóstrofes ni porcentajes”.</li> <li>5. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje de error de color rojo “Campo requerido” en el campo que debe ser llenado obligatorio.</li> <li>6. Si se introduce más caracteres de los permitidos para una palabra el sistema muestra el mensaje “Ha excedido el número de letras permitidas para una palabra”.</li> <li>7. En caso que se introduzca algún carácter que no sea número en el campo código de entidad o de organismo, el sistema muestra el mensaje de error: “Entre solo números”.</li> <li>8. En dependencia del organismo que se selecciona se muestra el código del organismo.</li> <li>9. Interactúa con esta acción el administrador del sistema, el Asesor Técnico Docente y la rectora.</li> </ol>	

Tabla 17: Descripción de requisito. Ver detalles entidad.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL10	Ver detalles entidad.	El requisito permite ver detalles de la entidad seleccionada por el usuario. Se muestran los datos en	Baja	Baja

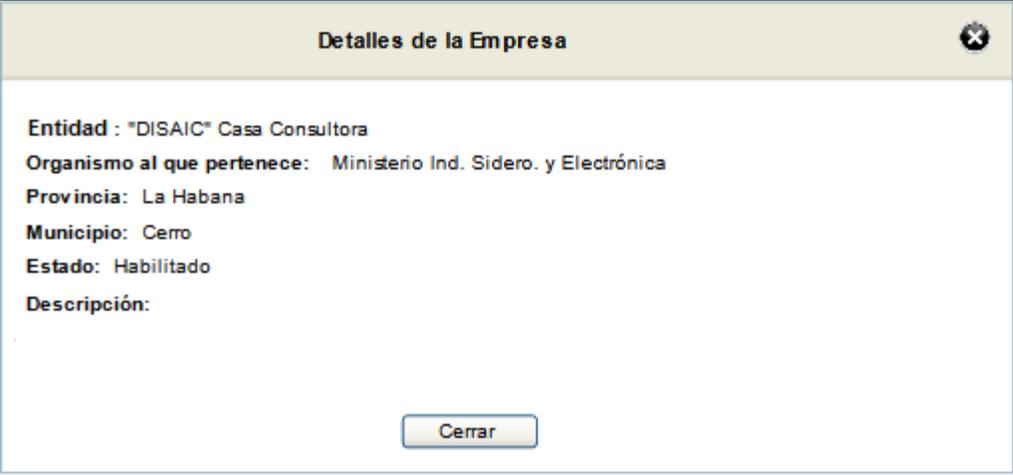
		<p>forma de ventana emergente.</p> <p>El requisito comienza cuando el administrador, el Asesor Técnico Docente o la rectora selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Comisiones</b> del menú de funcionalidades <b>Diseño de Ubicación Laboral</b>.</p>		
<b>Prototipo</b>				
 <p style="text-align: center;"><b>Prototipo Ver detalles de empresa.</b></p>				
	<b>Observaciones</b>	<p>1. Interactúa con esta acción el administrador del sistema, el Asesor Técnico Docente y la rectora.</p>		

Tabla 18: Descripción de requisito. Asociar registro de demanda.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL11	Asociar registro de demanda.	<p>El requisito permite asociar un registro de demanda a una entidad.</p> <p>El requisito comienza cuando el administrador, el Asesor Técnico Docente o la rectora selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Comisiones</b> del menú de funcionalidades <b>Diseño de Ubicación Laboral</b>.</p>	Alta	Alta

Se muestra un formulario a llenar con los campos Nombre (obligatorio), Cantidad de plazas (obligatorio), fecha (obligatorio), Descripción y Habilitado. En la parte inferior se muestra un listado con las demandas registradas a la entidad seleccionada.

**Prototipo**

**Registro de demandas**

Nombre:\*  Cantidad de plazas:\*  Descripción:

Fecha:\*

Habilitado

Asociar

Registro de demandas			
Nombre:	Cantidad de plazas:	Fecha:	
DISAIC Casa Consultora	40	11 00:00:00/03/2014	
DISAIC Casa Consultora	34	13 00:00:00/03/2014	
DISAIC Casa Consultora	10	27 00:00:00/03/2014	

Aceptar Cancelar

**Prototipo Asociar registro de demanda.**

Campos	Tipos de Datos	Reglas o Restricciones
Nombre	Character varying	Campo obligatorio. Admite a partir de 2 caracteres excepto % y comillas simples. La cantidad de caracteres permitidos para una palabra es de 30. La cantidad de caracteres permitidos es 250.
Cantidad de plazas	Integer	La cantidad de caracteres permitidos para un número es de 10.

	Descripción	Text	Admite a partir de 0 caracteres excepto % y comillas simples. La cantidad de caracteres permitidos para una palabra es de 30. La cantidad de caracteres permitidos es 1000.
	Fecha	Time with time zone	Permite seleccionar una fecha
	Habilitado	Boolean	Campo de selección que permite especificar si una demanda está habilitada o deshabilitada.
	<b>Observaciones</b>	<ol style="list-style-type: none"> <li>1. Si se crea satisfactoriamente el elemento se mostrará un mensaje de información “El elemento ha sido creado satisfactoriamente”.</li> <li>2. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de cancelar la acción?”. En caso que el usuario presione el botón Aceptar se regresa al listado de empresas. Si el usuario presiona el botón Cancelar se mantiene en la misma vista.</li> <li>3. En caso de que introduzca datos incorrectos (porcentaje % y comillas simples ‘) se muestra un mensaje de error “No se admiten apóstrofes ni porcentajes”.</li> <li>4. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje de error de color rojo “Campo requerido” en el campo que debe ser llenado obligatorio.</li> <li>5. Si se introduce más caracteres de los permitidos para una palabra el sistema muestra el mensaje “Ha excedido el número de letras permitidas para una palabra”.</li> <li>6. En caso que se introduzca algún carácter que no sea número en el campo código de entidad o de organismo, el sistema muestra el mensaje de error: “Entre solo números”.</li> <li>7. En el listado de la parte inferior solo permanecerá activo la última disponibilidad que se registro.</li> <li>8. Interactúa con esta acción el administrador del sistema, el Asesor Técnico</li> </ol>	

		Docente y la rectora.
--	--	-----------------------

Tabla 19: Descripción de requisito. Eliminar registro de demanda.

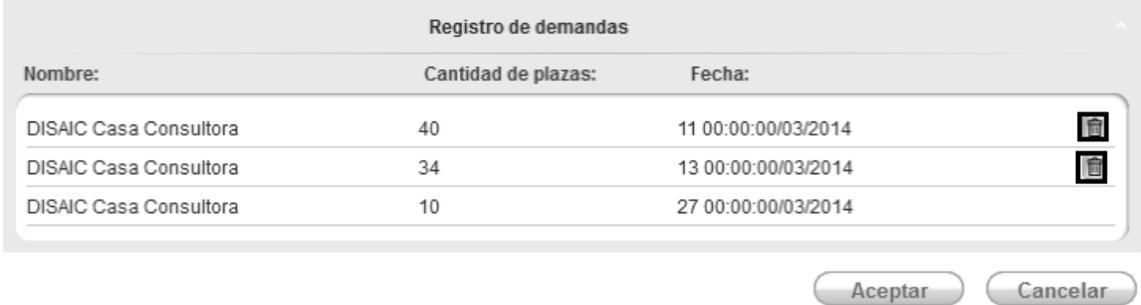
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL12	Eliminar registro de demanda.	<p>El requisito permite eliminar un registro de demanda a una entidad.</p> <p>El requisito comienza cuando el administrador, el Asesor Técnico Docente o la rectora selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Comisiones</b> del menú de funcionalidades <b>Diseño de Ubicación Laboral</b>.</p> <p>Se muestra un listado con las demandas registradas a la entidad seleccionada.</p>	Media	Media
<b>Prototipo</b>				
 <p style="text-align: center;"><b>Prototipo Eliminar registro de demanda.</b></p>				
	<b>Observaciones</b>	<ol style="list-style-type: none"> <li>1. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de cancelar la acción?”. En caso que el usuario presione el botón Aceptar se regresa al listado de empresas. Si el usuario presiona el botón Cancelar se mantiene en la misma vista.</li> <li>2. En el listado solo permanecerá activo la última disponibilidad que se registro.</li> <li>3. Interactúa con esta acción el administrador del sistema, el Asesor Técnico Docente y la rectora.</li> </ol>		

Tabla 20: Descripción de requisito. Mostrar facultades.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL13	Mostrar facultades.	<p>El requisito permite mostrar todas las facultades de la universidad.</p> <p>El requisito comienza cuando el administrador, Asesor Técnico Docente de la Oficina de Ubicación Laboral, el presidente de la comisión para la ubicación laboral a nivel de facultad o la rectora selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Índice de integralidad</b> del menú de funcionalidades <b>Ubicación Laboral</b>.</p> <p>Mostrándose un listado con las facultades de la universidad, y permitiendo seleccionar la opción mostrar indicadores de integralidad por cada facultad.</p>	Media	Media
<b>Prototipo</b>				
				

Prototipo Mostrar facultades.	
<b>Observaciones</b>	<ol style="list-style-type: none"> <li>1. Para realizar la acción el usuario tiene que estar autenticado en el sistema.</li> <li>2. La cantidad por página que se muestra en cada listado puede ser (5, 10, 15, 20).</li> <li>3. En el área de íconos flotantes del listado se encuentra el botón listar índice integralidad.</li> <li>4. Intervienen en esta acción el administrador del sistema, el Asesor Técnico Docente y la rectora.</li> </ol>

Tabla 21: Descripción de requisito. Mostrar índice de integralidad de estudiante.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL14	Mostrar indicadores de integralidad de estudiantes.	<p>El requisito permite mostrar los indicadores de integralidad de los estudiantes de un grupo administrativo seleccionado. El requisito comienza cuando el administrador, Asesor Técnico Docente de la Oficina de Ubicación Laboral, el presidente de la comisión para la ubicación laboral a nivel de facultad o la rectora selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Índice de integralidad</b> del menú de funcionalidades <b>Ubicación Laboral</b>.</p> <p>Mostrándose un listado con los datos: nombre y apellido del estudiante e indicadores de integralidad del estudiante, se mostrará los indicadores de integralidad en los casos en los que se encuentren registrados de lo contrario aparecerá el campo en blanco.</p> <p>Además se muestra la opción Registrar indicadores de integralidad, Modificar indicadores de integralidad para los estudiantes que tengan indicadores registrados, Ver Caracterización y ver los detalles de los estudiantes en el área de íconos internos del listado.</p>	Alta	Alta
<b>Prototipo</b>				

	 <p style="text-align: center;"><b>Prototipo Mostrar índice de integralidad de estudiante.</b></p>		
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>
	Grupo	Entero	<ul style="list-style-type: none"> <li>• Campo de selección.</li> </ul>
	<b>Observaciones</b>	<ol style="list-style-type: none"> <li>1. Para realizar la acción el usuario tiene que estar autenticado en el sistema.</li> <li>2. La cantidad por página que se muestra en cada listado puede ser (5, 10, 15, 20).</li> <li>3. En el área de íconos flotantes de la vista se encuentra el botón Listar por facultades y en el área de íconos flotantes se encuentran los botones Registrar índice integralidad, Modificar índice integralidad, Ver caracterización, para los estudiantes a los que se les haya adjuntado la caracterización de integralidad y ver detalles de los estudiantes.</li> <li>4. Intervienen en esta acción el administrador del sistema, el Asesor Técnico Docente y la rectora.</li> </ol>	

Tabla 22: Descripción de requisito. Registrar indicadores de integralidad a estudiante.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL15	Registrar indicadores de integralidad a estudiante.	El requisito permite introducir cada uno de los valores de los indicadores de integralidad a un estudiante seleccionado. El requisito comienza cuando el administrador, el	Media	Alta

		<p>Asesor Técnico Docente de la Oficina de Ubicación Laboral ó presidente de la comisión para la ubicación laboral a nivel de facultad selecciona la acción <b>registrar indicadores de integralidad</b> del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Índice de integralidad</b> del menú de funcionalidades <b>Ubicación Laboral</b>.</p> <p>Se muestra un formulario con los indicadores de integralidad:</p> <ul style="list-style-type: none"> <li>• Producción (obligatorio)</li> <li>• FEU-UJC (obligatorio)</li> <li>• Investigación (obligatorio)</li> <li>• TSU (obligatorio)</li> <li>• Deporte (obligatorio)</li> <li>• Guardia (obligatorio)</li> <li>• Cultura (obligatorio)</li> <li>• Residencia (obligatorio)</li> <li>• Premio Mella</li> <li>• Título de Oro</li> <li>• Caracterización</li> <li>• Medidas disciplinarias menos graves</li> <li>• Medidas disciplinarias graves</li> <li>• Medidas disciplinarias menos graves</li> <li>• Arrastres</li> <li>• Mundiales</li> <li>• Repitentes</li> </ul> <p>Los cuales el usuario debe llenar.</p> <p>Si al presionar el botón Aceptar existen datos que no cumplan con el formato designado para cada campo el sistema muestra un mensaje de error especificando porqué es incorrecto el dato introducido, no pudiéndose realizar la operación</p>		
--	--	--	--	--

hasta que no se corrija el error.

**Prototipo**

**Índices de estudiante**

GENERALES

Nombre y Apellidos: Ana Belkis Granados Arozarena

Promedio Académico: 4.409

Producción:\*

FEU-UJC:\*

Investigación:\*

Extensión

TSU:\*

Deporte:\*

Guardia:\*

Cultura:\*

Residencia:\*

Reconocimientos

Premio Mella:

Título de Oro:

Caracterización: pdf

Agravantes

Medidas disciplinarias menos graves:

Medidas disciplinarias graves:

Medidas disciplinarias muy graves:

Arrastres:

Mundiales:

Repitentes:

**Prototipo Registrar indicadores de integralidad a estudiante.**

Campos	Tipos de Datos	Reglas o Restricciones
Producción	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
FEU-UJC	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
Investigación	Integer	Obligatorio.

			Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
	TSU	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
	Deporte	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
	Guardia	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
	Cultura	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
	Residencia	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
	Premio Mella	Boolean	-
	Título de Oro	Boolean	-

	Caracterización	File	Permite adjuntar un fichero con extensión .pdf.
	Medidas disciplinarias menos graves	Integer	Permite introducir un valor mayor que cero y menor que 50, incluidos ambos.
	Medidas disciplinarias graves	Integer	Permite introducir un valor mayor que cero y menor que 50, incluidos ambos.
	Medidas disciplinarias graves	Integer	Permite introducir un valor mayor que cero y menor que 50, incluidos ambos.
	Repitente	Boolean	-
	<b>Observaciones</b>	<ol style="list-style-type: none"> <li>1. Si se da click en el botón aceptar y hay campos obligatorios sin llenar el sistema mostrara un mensaje sobre el campo: "Campo obligatorio".</li> <li>2. Si se da click en el botón aceptar y hay campos con sus valores en un formato fuera del establecido el sistema mostrara un mensaje sobre el campo indicando lo incorrecto de este valor.</li> <li>3. Si el usuario presiona el botón Cancelar se muestra el mensaje "¿Está seguro de realizar la acción?".</li> <li>4. Para realizar la acción el usuario tiene que estar autenticado en el sistema.</li> <li>5. La cantidad por página que se muestra en cada listado puede ser (5, 10, 15, 20).</li> <li>6. Intervienen en esta acción el administrador del sistema, el Asesor Técnico Docente y la rectora.</li> </ol>	

Tabla 23: Descripción de requisito. Modificar indicadores de integralidad a estudiante.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL16	Modificar indicadores de integralidad a estudiante.	El requisito permite modificar cada uno de los valores de los indicadores de integralidad a un estudiante seleccionado. El requisito comienza cuando el administrador, el	Media	Media

		<p>Asesor Técnico Docente de la Oficina de Ubicación Laboral ó presidente de la comisión para la ubicación laboral a nivel de facultad selecciona la acción <b>modificar indicadores de integralidad</b> del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Índice de integralidad</b> del menú de funcionalidades <b>Ubicación Laboral</b>.</p> <p>Se muestra un formulario con los indicadores de integralidad:</p> <ul style="list-style-type: none"> <li>• Producción (obligatorio)</li> <li>• FEU-UJC (obligatorio)</li> <li>• Investigación (obligatorio)</li> <li>• TSU (obligatorio)</li> <li>• Deporte (obligatorio)</li> <li>• Guardia (obligatorio)</li> <li>• Cultura (obligatorio)</li> <li>• Residencia (obligatorio)</li> <li>• Premio Mella</li> <li>• Título de Oro</li> <li>• Caracterización</li> <li>• Medidas disciplinarias menos graves</li> <li>• Medidas disciplinarias graves</li> <li>• Medidas disciplinarias menos graves</li> <li>• Arrastres</li> <li>• Mundiales</li> <li>• Repitentes</li> </ul> <p>Los cuales contienen los indicadores de integralidad registrados al estudiante seleccionado y los cuales el usuario puede modificar.</p> <p>Si al presionar el botón Aceptar existen datos que no cumplan con el formato designado para cada campo el sistema muestra un mensaje de error</p>	
--	--	--	--

especificando porqué es incorrecto el dato introducido, no pudiéndose realizar la operación hasta que no se corrija el error.

**Prototipo**

**Modificar índices de estudiante**

**Prototipo Modificar indicadores de integralidad a estudiante.**

Campos	Tipos de Datos	Reglas o Restricciones
Producción	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
FEU-UJC	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
Investigación	Integer	Obligatorio. Permite seleccionar un

			número mayor que cero y menor que cinco, incluidos ambos.
	TSU	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
	Deporte	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
	Guardia	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
	Cultura	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
	Residencia	Integer	Obligatorio. Permite seleccionar un número mayor que cero y menor que cinco, incluidos ambos.
	Premio Mella	Boolean	-
	Título de Oro	Boolean	-
	Caracterización	File	Permite adjuntar un fichero con extensión .pdf.

	Medidas disciplinarias menos graves	Integer	Permite introducir un valor mayor que cero y menor que 50, incluidos ambos.
	Medidas disciplinarias graves	Integer	Permite introducir un valor mayor que cero y menor que 50, incluidos ambos.
	Medidas disciplinarias graves	Integer	Permite introducir un valor mayor que cero y menor que 50, incluidos ambos.
	Repitente	Boolean	=
	<b>Observaciones</b>	<ol style="list-style-type: none"> <li>1. Si se da click en el botón aceptar y hay campos obligatorios sin llenar el sistema mostrara un mensaje sobre el campo: "Campo obligatorio".</li> <li>2. Si se da click en el botón aceptar y hay campos con sus valores en un formato fuera del establecido el sistema mostrara un mensaje sobre el campo indicando lo incorrecto de este valor.</li> <li>3. Si el usuario presiona el botón Cancelar se muestra el mensaje "¿Está seguro de realizar la acción?".</li> <li>4. Para realizar la acción el usuario tiene que estar autenticado en el sistema.</li> <li>5. La cantidad por página que se muestra en cada listado puede ser (5, 10, 15, 20).</li> <li>6. Intervienen en esta acción el administrador del sistema, el Asesor Técnico Docente y la rectora.</li> </ol>	

Tabla 24: Descripción de requisito. Ver detalles indicadores integralidad a estudiantes.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL17	Ver detalles indicadores integralidad a estudiantes.	El requisito permite ver los detalles de indicadores integralidad a estudiantes. El requisito comienza cuando el administrador, el Asesor Técnico Docente de la Oficina de Ubicación Laboral ó presidente de la comisión para la ubicación laboral a nivel de facultad selecciona la acción	Baja	Baja

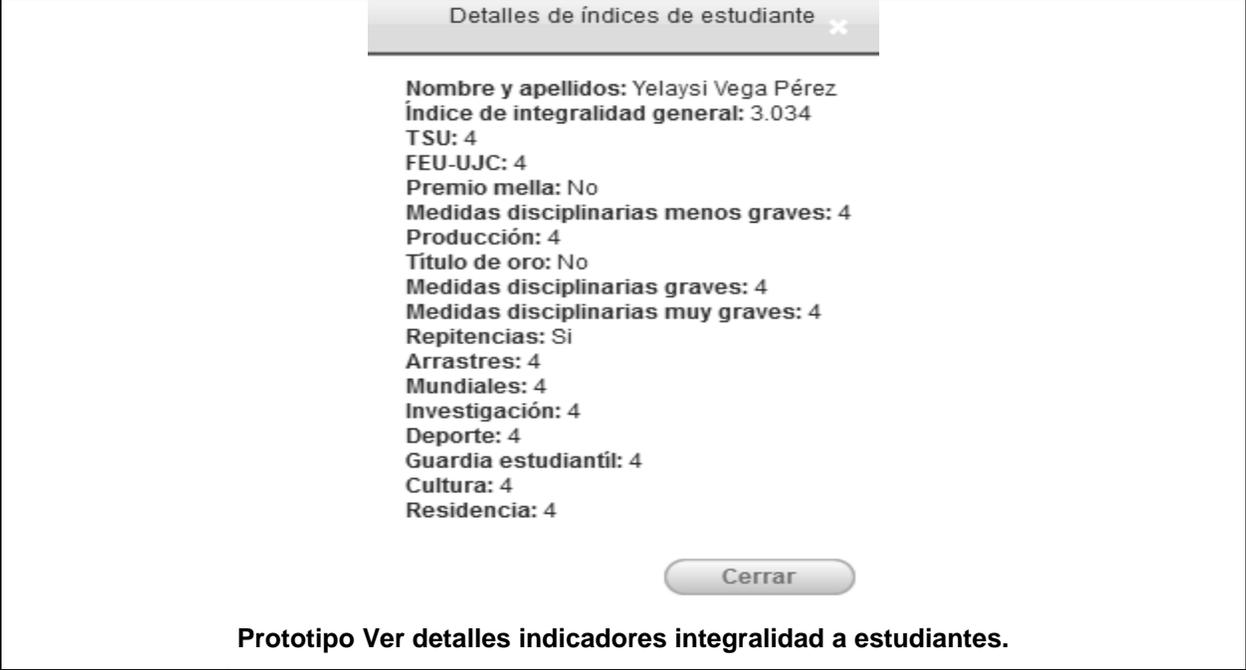
		<p><b>modificar indicadores de integralidad</b> del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Índice de integralidad</b> del menú de funcionalidades <b>Ubicación Laboral</b>.</p>		
<b>Prototipo</b>				
 <p style="text-align: center;"><b>Prototipo Ver detalles indicadores integralidad a estudiantes.</b></p>				
	<b>Observaciones</b>	<p>1. Intervienen en esta acción el administrador del sistema, el Asesor Técnico Docente o la rectora.</p>		

Tabla 25: Descripción de requisito. Mostrar caracterización de estudiante.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL18	Mostrar caracterización de estudiante.	<p>El requisito permite Permite mostrar la caracterización de estudiante.</p> <p>El requisito comienza cuando el administrador, el Asesor Técnico Docente de la Oficina de Ubicación Laboral ó presidente de la comisión para la ubicación laboral a nivel de facultad selecciona la acción <b>Mostrar caracterización</b> del módulo <b>Ubicación</b></p>	Baja	Alta

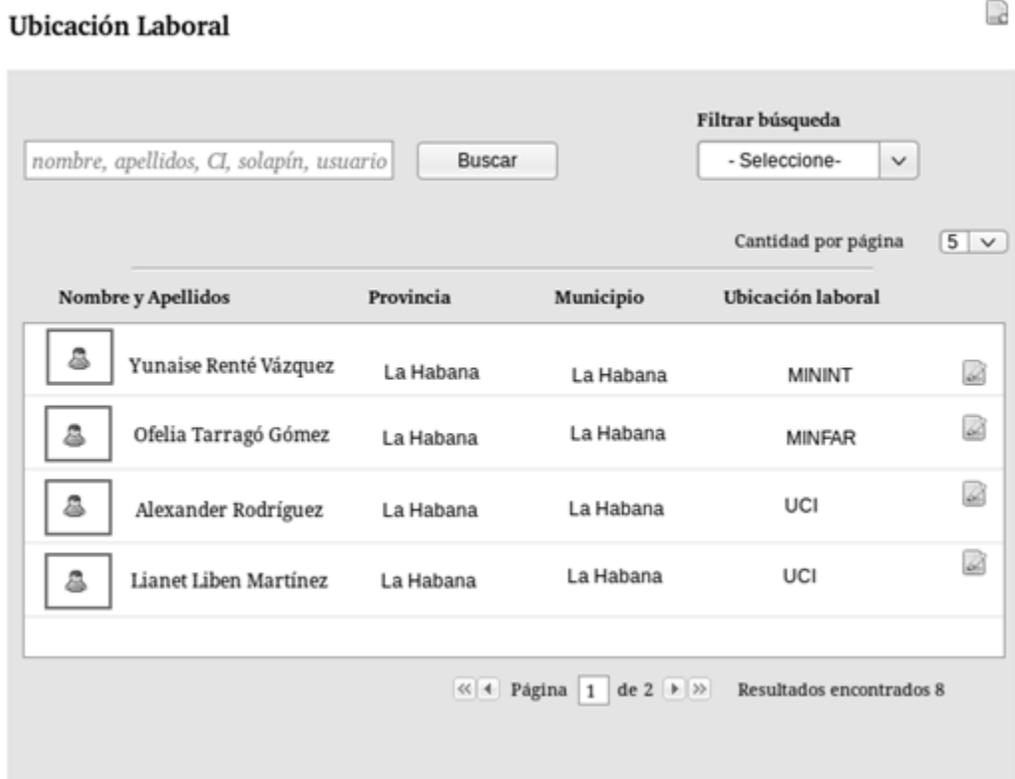
		<b>Laboral</b> del subsistema <b>Egreso</b> , la opción <b>Índice de integralidad</b> del menú de funcionalidades <b>Ubicación Laboral</b> . Especificar la forma en que se termina la acción.		
<b>Prototipo</b>				
 <p style="text-align: center;"><b>Prototipo Mostrar caracterización de estudiante.</b></p>				
	<b>Observaciones</b>	<ol style="list-style-type: none"> <li>1. Solo aparecerá el icono de ver caracterización a solo aquellos estudiantes a los que se le haya adjuntado el acta de caracterización.</li> <li>2. Intervienen en esta acción el administrador del sistema, el Asesor Técnico Docente y la rectora.</li> </ol>		

Tabla 26: Descripción de requisito. Mostrar ubicación laboral de estudiante.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL19	Mostrar ubicación laboral.	El requisito permite mostrar todas las ubicaciones laborales que se han asignado a los estudiantes. El requisito comienza cuando el Asesor Técnico Docente de la Oficina de Ubicación Laboral selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b> , la opción <b>Ubicación laboral</b> del	Media	Alta

menú de funcionalidades **Ubicación Laboral**.  
 Mostrándose un listado con los datos: nombre y apellido del estudiante, provincia y municipio del estudiante, y plaza al que fue ubicado. Además se muestra la opción Modificar plaza en el área de íconos interna del listado.

**Prototipo**



**Prototipo Mostrar ubicación laboral.**

**Observaciones**

1. Para realizar la acción el usuario tiene que estar autenticado en el sistema.
2. La cantidad por página que se muestra en cada listado puede ser (5, 10, 15, 20).
3. En el área de íconos flotantes de la vista se encuentra la funcionalidad Asignar ubicación laboral y en el área de íconos flotantes se encuentra la funcionalidad Modificar plaza.

		<p>4. En el listado solo se mostrarán los estudiantes que tengan plazas asignadas.</p> <p>5. Tienen acceso a esta funcionalidad el administrador del sistema, el Asesor Técnico Docente y la rectora.</p>
--	--	---

Tabla 27: Descripción de requisito. Modificar ubicación laboral de estudiante.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL21	Modificar ubicación laboral a estudiante.	<p>El requisito permite modificar la plaza laboral asignada a un estudiante.</p> <p>El requisito comienza cuando el administrador del sistema, Asesor Técnico Docente de la Oficina de Ubicación Laboral o la rectora selecciona del módulo <b>Ubicación Laboral</b> del subsistema <b>Egreso</b>, la opción <b>Ubicación laboral</b> del menú de funcionalidades <b>Ubicación Laboral</b>.</p> <ol style="list-style-type: none"> <li>Mostrándose un listado con los estudiantes que están ubicados en plaza.</li> <li>El usuario selecciona la opción Modificar del estudiante que desee modificar su plaza.</li> <li>Se muestra una vista con los datos <b>personales del estudiante</b> (nombre y apellido, provincia, municipio, facultad, índice general) y los datos de ubicación laboral (entidad, organismo, provincia y municipio).</li> <li>Además se muestra un listado con todas empresas que tienen plazas disponibles y al que se le puede asociar estudiantes.</li> <li>El usuario selecciona la nueva plaza que desee y presiona el botón Aceptar, mostrándose un mensaje: "La acción se ha realizado satisfactoriamente".</li> </ol> <p>De esta forma al estudiante se le elimina la plaza que tenía anteriormente y se le asigna la nueva</p>	Media	Alta

plaza.

## Prototipo

### Modificar ubicación laboral



Foto

Nombre: Diana Galardy Tarragó  
Provincia: Granma  
Municipio: Jiguani  
Facultad: Facultad 1

Entidad: UCI  
Organismo: MES  
Provincia: La Habana  
Municipio: La Lisa

nombre,

Buscar

Filtrar búsqueda:

-Selecione-

#### Entidades

Cantidad por página

5

	Prioridad	Entidad	Provincia	Municipio	Diponibilidad
<input type="radio"/>	1	UCI	La Habana	La Habana	1
<input type="radio"/>	2	MININT	La Habana	La Habana	1
<input type="radio"/>	3	Emp. Agropecuaria	Matanzas	Matanzas	1
<input type="radio"/>	4	Emp. Petróleo	Holguín	Holguín	1

<< Página 1 de 2 >> Resultados encontrados 8

Aceptar

Cancelar

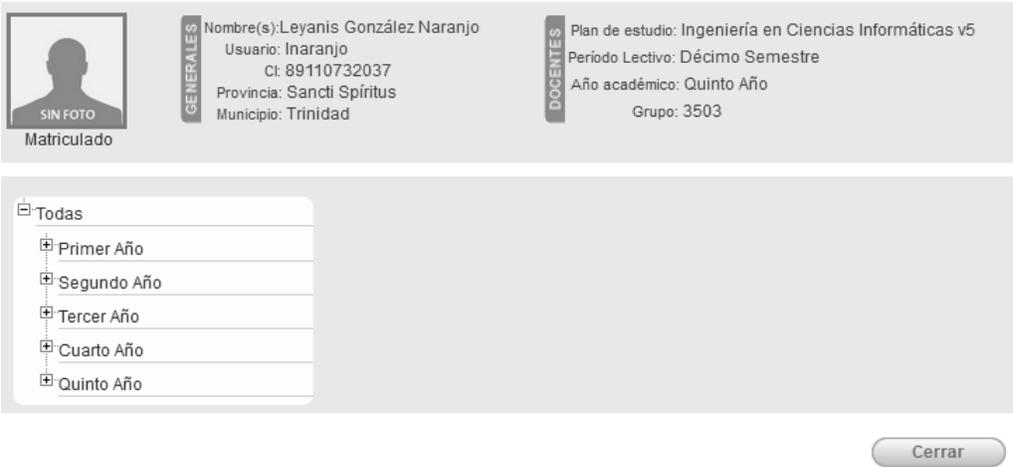
### Prototipo Modificar ubicación laboral a estudiante.

#### Observaciones

1. Para realizar la acción el usuario tiene que estar autenticado en el sistema.
2. En el listado solo se muestran las empresas que tengan disponibilidad de plazas.
3. Cada vez que se asocie una plaza a estudiante se debe disminuir la disponibilidad de plaza para la entidad seleccionada.
4. Si el usuario presiona el botón Cancelar se muestra el mensaje "¿Está seguro de realizar la acción?", si presiona Aceptar se regresa al listado de Mostrar

		<p>asignación de plazas. Si presiona el botón Cancelar se mantiene en la página sin realizar ninguna operación.</p> <p>5. La cantidad por página que se muestra en cada listado puede ser (5, 10, 15, 20).</p> <p>6. Intervienen en esta acción el administrador del sistema, el Asesor Técnico Docente y la rectora.</p>
--	--	---

Tabla 28: Descripción de requisito. Mostrar resumen de evaluaciones.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFUL22	Mostrar resumen de evaluaciones.	El requisito permite mostrar el resumen de evaluaciones de toda la carrera de un estudiante. El administrador del sistema, el Asesor Técnico Docente o la rectora selecciona del módulo <b>Ubicación Laboral</b> del subsistema de <b>Egreso</b> , la opción <b>Asignar plaza a estudiante</b> del menú de funcionalidades <b>Ubicación Laboral</b> .	Alta	Alta
<b>Prototipo</b>				
<p><b>Resumen de evaluaciones</b></p>  <p><b>Prototipo Mostrar resumen de evaluaciones.</b></p>				
	<b>Observaciones</b>	1. Intervienen en esta acción el administrador del sistema, el Asesor Técnico Docente y la rectora.		

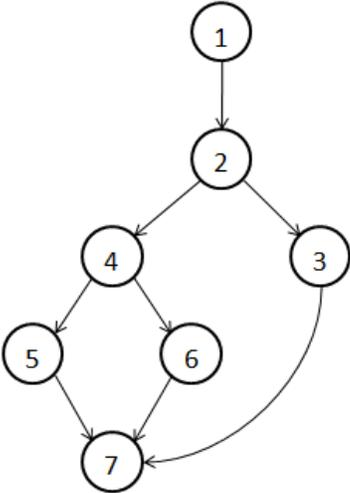
#### Anexo 4: Criterios para validar requerimientos del cliente

Tabla 29: Criterios para validar requerimientos del cliente.

<b>Criterios para validar requerimientos del cliente</b>
¿El proveedor del requisito es un proveedor válido?
¿El requisito tiene un identificador único?
¿El requisito es modificable?
¿El requisito no es ambiguo?
¿El requisito está completo?
¿El requisito es congruente con otros requisitos relacionados?
¿El requisito puede ser implementado?
¿El requisito puede ser probado?
¿El resultado de la evaluación de impacto es positivo?
¿El requisito es correcto?
¿El requisito es traceable?

## Anexo 5: Casos de prueba de caja blanca

Tabla 30: Caso de prueba de caja blanca del método chequearNombreRepetidoModificar.

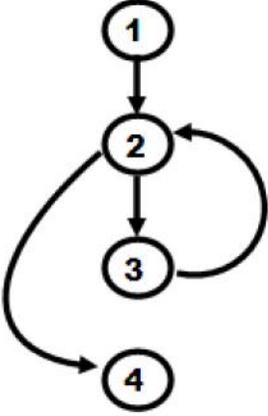
<b>Prueba estructural de caja blanca</b>	<b>Código caso de prueba: DIN_Egre_UL_OCP</b>
<b>Probador:</b> Froilán Morales Medina	
<b>Código al que se aplica:</b> <pre> public function chequearNombreRepetidoModificar(\$post_vars) {      \$arreglo_ids = array("nombre_organismo" =&gt; \$post_vars['nombre_organismo'],                         "codigo_organismo" =&gt; \$post_vars['codigo_organismo']);     \$organismo = \$this-&gt;_ci-&gt;tb_norganismo_mdl-&gt;obtenerDadoIdModificar(\$arreglo_ids);     if (!empty(\$organismo)) {         if (\$organismo-&gt;id_organismo != \$post_vars['id_organismo'])             return true;         else             return false;     } else         return false; }         </pre>	
<b>Complejidad ciclomática:</b> $V(G) = A - N + 2$ $V(G) = 8 - 7 + 2 = 3$  <b>Caminos independientes:</b> 1-2-4-5-7 1-2-4-6-7 1-2-3-7	<b>Representación del grafo:</b> 
<b>Caso de prueba para el camino básico 1</b>	
<b>Descripción:</b> los datos de entrada serán los atributos de un organismo que coinciden con los atributos de otro	

organismo registrado en la Base de Datos.	
<b>Condición de ejecución:</b> el usuario debe estar autenticado y tener permiso sobre la funcionalidad crear organismo.	
<b>Procedimiento prueba automatizada</b>	
<b>Datos de entrada:</b>	Un arreglo con los datos de un organismo a modificar.
<b>Tipo de dato esperado:</b>	True.
<b>Función de evaluación:</b>	
<pre>\$nombrePrueba="Prueba 1: Atributos de un organismos repetidos en otros organismos"; \$resultadoEsperado=true; pr( \$this-&gt;_ci-&gt;unit-&gt;run(\$this-&gt;chequearNombreRepetidoModificar(\$datos), \$resultadoEsperado, \$nombrePrueba);</pre>	
<b>Evaluación del caso de prueba:</b>	
Test Name	Prueba 1: Atributos de un organismos repetidos en otros organismos
Test Datatype	Boolean
Expected Datatype	Boolean
Result	Passed
File Name	C:\wamp\www\nucleo\base\application\egreso\ubicacion_laboral\libraries\organismo_lib.php
Line Number	110
<b>Caso de prueba para el camino básico 2</b>	
<b>Descripción:</b> los datos de entrada serán los atributos del organismo que se quiere modificar y no se encuentra otro organismo con esos atributos en la Base de Datos.	
<b>Condición de ejecución:</b> el usuario debe estar autenticado y tener permiso sobre la funcionalidad crear organismo.	
<b>Procedimiento prueba automatizada</b>	
<b>Datos de entrada:</b>	Un arreglo con los datos de un organismo.
<b>Tipo de dato esperado:</b>	False.
<b>Función de evaluación:</b>	
<pre>\$nombrePrueba="Prueba 2: Atributos del organismos que se quiere modificar y no se encuentran repetidos en otros organismos"; \$resultadoEsperado=false; pr( \$this-&gt;_ci-&gt;unit-&gt;run(\$this-&gt;chequearNombreRepetidoModificar(\$datos), \$resultadoEsperado, \$nombrePrueba);</pre>	
<b>Evaluación del caso de prueba:</b>	
Test Name	Prueba 2: Atributos del organismos que se quiere modificar y no se encuentran repetidos en otros organismos
Test Datatype	Boolean
Expected Datatype	Boolean
Result	Passed
File Name	C:\wamp\www\nucleo\base\application\egreso\ubicacion_laboral\libraries\organismo_lib.php
Line Number	110
<b>Caso de prueba para el camino básico 3</b>	
<b>Descripción:</b> los datos de entrada serán los atributos del organismo que se quiere modificar y no se encuentra otro	

organismo con esos atributos en la Base de Datos. Ni en el propio organismo.	
<b>Condición de ejecución:</b> el usuario debe estar autenticado y tener permiso sobre la funcionalidad crear organismo.	
<b>Procedimiento prueba automatizada</b>	
<b>Datos de entrada:</b>	Un arreglo con los datos de un organismo.
<b>Tipo de dato esperado:</b>	False.
<b>Función de evaluación:</b>	
<pre> \$nombrePrueba="Prueba 3: Atributos del organismos que se quiere modificar y no se encuentran repetidos en otros organismos, ni en propio organismo"; \$resultadoEsperado=false; pr( \$this-&gt;_ci-&gt;unit-&gt;run(\$this-&gt;chequearNombreRepetidoModificar(\$datos), \$resultadoEsperado, \$nombrePrueba)); </pre>	
<b>Evaluación del caso de prueba:</b>	
Test Name	Prueba 3: Atributos del organismos que se quiere modificar y no se encuentran repetidos en otros organismos, ni en propio organismo
Test Datatype	Boolean
Expected Datatype	Boolean
Result	Passed
File Name	C:\wamp\www\nucleo\base\application\egreso\ubicacion_laboral\libraries\organismo_lib.php
Line Number	110
<b>Resultado final de la prueba:</b> satisfactoria en su totalidad.	

Tabla 31: Caso de prueba de caja blanca del método asociarRolPersona.

<b>Prueba estructural de caja blanca</b>	<b>Código caso de prueba: DIN_Egre_UL_ARP</b>
<b>Probador:</b> Froilán Morales Medina	
<b>Código al que se aplica:</b>	
<pre> public function asociarRolPersona(\$datos) {      \$datos_finales = array();     \$id_comision['id_comision'] = \$datos['id_comision'];     \$asociados = json_decode(\$datos['asociados']);     \$this-&gt;_ci-&gt;tb_rpersona_comision_rol_md1-&gt;eliminarDadoId(array('id_comision' =&gt; \$id_comision['id_comision']));     foreach (\$asociados as \$llave =&gt; \$valor) {         \$datos_finales['id_persona'] = \$valor-&gt;id_persona;         \$datos_finales['id_rol_comision'] = \$valor-&gt;id_rol_comision;         \$datos_finales['id_comision'] = \$id_comision['id_comision'];         \$this-&gt;_ci-&gt;tb_rpersona_comision_rol_md1-&gt;registrar(\$datos_finales);     }     return true; } </pre>	
<b>Complejidad ciclomática:</b>	<b>Representación del grafo:</b>
$V(G) = A - N + 2$ $V(G) = 4 - 4 + 2 = 2$	

<p><b>Caminos independientes:</b></p> <p>1-2-4</p> <p>1-2-3-2-4</p>	 <pre> graph TD     1((1)) --&gt; 2((2))     2 --&gt; 3((3))     3 --&gt; 2     2 --&gt; 4((4)) </pre>
---	--

**Caso de prueba para el camino básico 1**

**Descripción:** los datos de entrada serán el identificador de comisión y los datos de las personas a asociar.

**Condición de ejecución:** el usuario debe estar autenticado y tener permiso sobre la funcionalidad asociar registro de demanda.

**Procedimiento prueba automatizada**

**Datos de entrada:** Un arreglo con el identificador de comisión y los datos de las personas a asociar.

**Tipo de dato esperado:** True.

**Función de evaluación:**

```

$nombrePrueba="Prueba 1: Asociar rol de persona sin datos de personas";
$resultadoEsperado=true;
pr( $this->_ci->unit->run($this->asociarRolPersona($datos), $resultadoEsperado, $nombrePrueba) );

```

**Evaluación del caso de prueba:**

Test Name	Prueba 1: Asociar rol de persona sin datos de personas
Test Datatype	Boolean
Expected Datatype	Boolean
Result	Passed
File Name	C:\wamp\www\nucleo\base\application\egreso\ubicacion_laboral\libraries\comision_lib.php
Line Number	159

**Caso de prueba para el camino básico 2**

**Descripción:** los datos de entrada serán el identificador de comisión y los datos de las personas a asociar

**Condición de ejecución:** el usuario debe estar autenticado y tener permiso sobre la funcionalidad asociar registro de demanda.

**Procedimiento prueba automatizada**

<b>Datos de entrada:</b>	Un arreglo con el identificador de comisión y los datos de las personas a asociar.
<b>Tipo de dato esperado:</b>	True.
<b>Función de evaluación:</b>	
<pre> \$nombrePrueba="Prueba 2: Asociar rol de persona con datos de personas"; \$resultadoEsperado=true; pr( \$this-&gt;_ci-&gt;unit-&gt;run(\$this-&gt;asociarRolPersona(\$datos), \$resultadoEsperado, \$nombrePrueba) ); </pre>	
<b>Evaluación del caso de prueba:</b>	
Test Name	Prueba 2: Asociar rol de persona con datos de personas
Test Datatype	Boolean
Expected Datatype	Boolean
Result	Passed
File Name	C:\wamp\www\nucleo\base\application\egreso\ubicacion_laboral\libraries\comision_lib.php
Line Number	159
<b>Resultado final de la prueba:</b> satisfactoria en su totalidad.	

## Anexo 6: Pruebas de integración

Tabla 32: Prueba de integración entre las funcionalidades registrar indicadores de integralidad a un estudiante y obtener promedio de carrera.

<b>N° de caso de prueba:</b> Int_UL_2	
<b>Funcionalidad a integrar:</b>	Registrar indicadores de integralidad a estudiante.
<b>Módulo al que se integra:</b>	Control Docente.
<b>Condiciones de ejecución:</b>	El usuario debe estar autenticado en el sistema. El módulo Control Docente haya introducido los datos en la base de datos central y exista conexión con la misma.
<b>Descripción de la prueba:</b>	Comprobar que el módulo de Ubicación Laboral sea capaz de obtener el promedio académico para registrar el índice de integralidad a un estudiante con información gestionada por el módulo Control Docente.
<b>Entrada/Pasos de ejecución:</b>	El usuario se dirige a la agrupación funcional Ubicación Laboral, accede a la funcionalidad indicadores de integralidad, luego selecciona una facultad y un grupo docente. Después selecciona la opción registrar indicadores de integralidad a un estudiante del menú de íconos internos.
<b>Resultado esperado:</b>	Se logra registrar el índice de integralidad a un estudiante correctamente.
<b>Evaluación:</b>	Prueba satisfactoria.

Tabla 33: Prueba de integración entre las funcionalidades asignar ubicación laboral a estudiante y obtener estudiantes foto buscador.

<b>N° de caso de prueba:</b> Int_UL_3	
<b>Funcionalidad a integrar:</b>	Asignar ubicación laboral a estudiante.
<b>Módulo al que se integra:</b>	Personal Secretaria.
<b>Condiciones de ejecución:</b>	El usuario debe estar autenticado en el sistema. El módulo Personal Secretaria haya introducido los datos en la base de datos central y exista conexión con la misma.
<b>Descripción de la prueba:</b>	Comprobar que el módulo de Ubicación Laboral sea capaz de obtener la información referente a las fotos de los estudiantes gestionada por el módulo Personal Secretaria.
<b>Entrada/Pasos de ejecución:</b>	El usuario se dirige a la agrupación funcional Ubicación Laboral, accede a la funcionalidad Asignar plaza a estudiante, luego selecciona la opción

	Asignar plaza del menú de íconos flotantes.
<b>Resultado esperado:</b>	Se logra obtener la información referente a las fotos de los estudiantes correctamente.
<b>Evaluación:</b>	Prueba satisfactoria.

Tabla 34: Prueba de integración entre las funcionalidades asignar prioridad a entidad y obtener división administrativa.

<b>N° de caso de prueba: Int_UL_4</b>	
<b>Funcionalidad a integrar:</b>	Asignar prioridad a entidad.
<b>Módulo al que se integra:</b>	Configuración.
<b>Condiciones de ejecución:</b>	El usuario debe estar autenticado en el sistema. El módulo Configuración haya introducido los datos en la base de datos central y exista conexión con la misma.
<b>Descripción de la prueba:</b>	Comprobar que el módulo de Ubicación Laboral sea capaz de obtener la información referente a la división administrativa del país gestionada por el módulo Configuración.
<b>Entrada/Pasos de ejecución:</b>	El usuario se dirige a la agrupación funcional Ubicación Laboral, accede a la funcionalidad Configuración, luego selecciona la opción Prioridad de entidad del menú de íconos flotantes.
<b>Resultado esperado:</b>	Se logra obtener la información referente a la división administrativa del país correctamente.
<b>Evaluación:</b>	Prueba satisfactoria.

Tabla 35: Prueba de integración entre las funcionalidades asignar estudiantes a grupo de certificación y obtener elementos dado token.

<b>N° de caso de prueba: Int_UL_5</b>	
<b>Funcionalidad a integrar:</b>	Asignar estudiantes a grupo de certificación
<b>Módulo al que se integra:</b>	Seguridad
<b>Condiciones de ejecución:</b>	El usuario debe estar autenticado en el sistema.
<b>Descripción de la prueba:</b>	Comprobar que el módulo de Ubicación Laboral sea capaz de obtener la información del usuario autenticado gestionada por el módulo Seguridad.

<b>Entrada/Pasos de ejecución:</b>	El usuario se dirige a la agrupación funcional Boletas, accede a la funcionalidad Grupo de certificación, luego selecciona la opción Asignar estudiantes del menú de íconos internos.
<b>Resultado esperado:</b>	Se logra obtener la información del usuario autenticado en el sistema correctamente.
<b>Evaluación:</b>	Prueba satisfactoria.

## Anexo 7: Pruebas de rendimiento

A continuación se muestran los resultados de las pruebas de rendimientos realizadas al módulo de Ubicación Laboral para 1 y 50 usuarios conectados concurrentemente.

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/sgu/base/ubica...	1	453	453	453	453	453	0.00%	2.2/sec	1.1
/sgu/base/ubica...	1	406	406	406	406	406	0.00%	2.5/sec	2
/sgu/base/ubica...	1	266	266	266	266	266	0.00%	3.8/sec	2.0
/sgu/base/ubica...	1	313	313	313	313	313	0.00%	3.2/sec	2
/sgu/base/ubica...	1	406	406	406	406	406	0.00%	2.5/sec	2.3
TOTAL	5	368	406	453	266	453	0.00%	2.7/sec	1.1

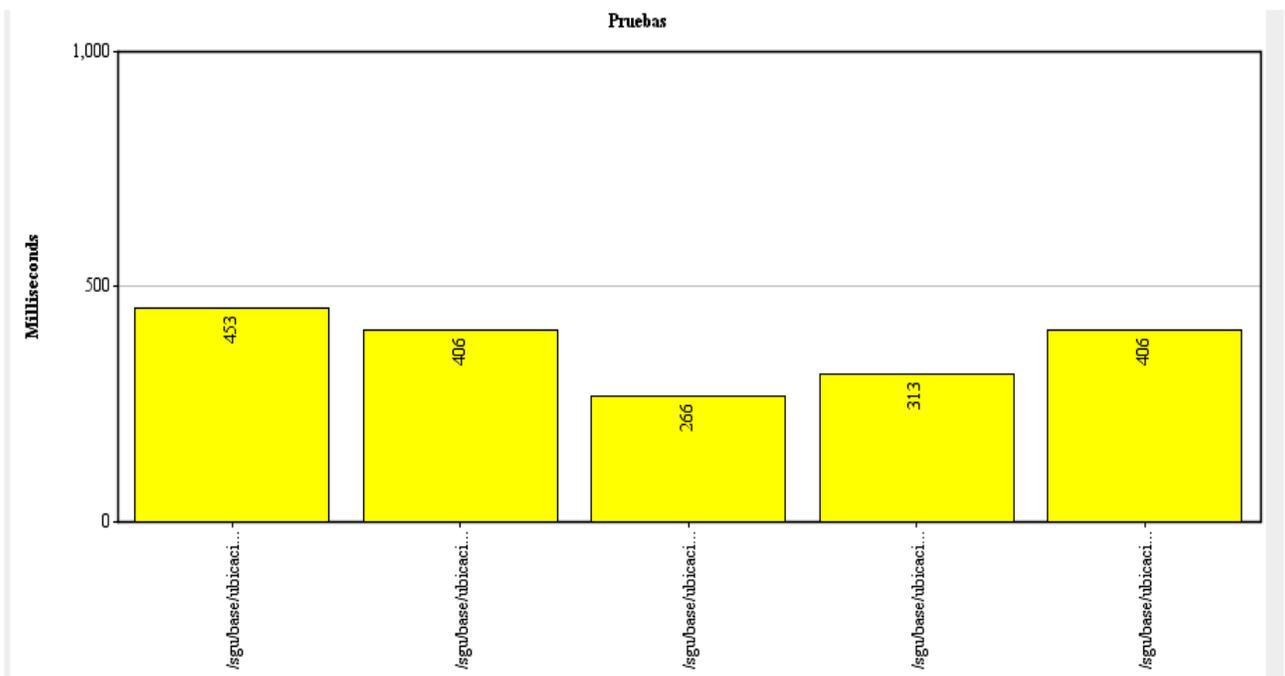


Figura 20: Resultado de las pruebas de rendimiento para 1 usuario.

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/sgu/base/ubic...	50	2204	1804	5798	339	6278	0,00%	1,8/sec	,4
/sgu/base/ubic...	50	1771	1810	2607	594	6293	0,00%	1,7/sec	1,6
/sgu/base/ubic...	50	2665	2755	3606	654	7177	0,00%	1,6/sec	18,3
/sgu/base/ubic...	50	2996	3123	3819	1142	7245	0,00%	1,5/sec	1,2
/sgu/base/ubic...	50	2777	2933	3891	933	7328	0,00%	1,5/sec	4,8
/sgu/base/ubic...	50	2660	3013	3662	958	6914	0,00%	1,5/sec	,1
TOTAL	300	2512	2539	3771	339	7328	0,00%	8,2/sec	22,9

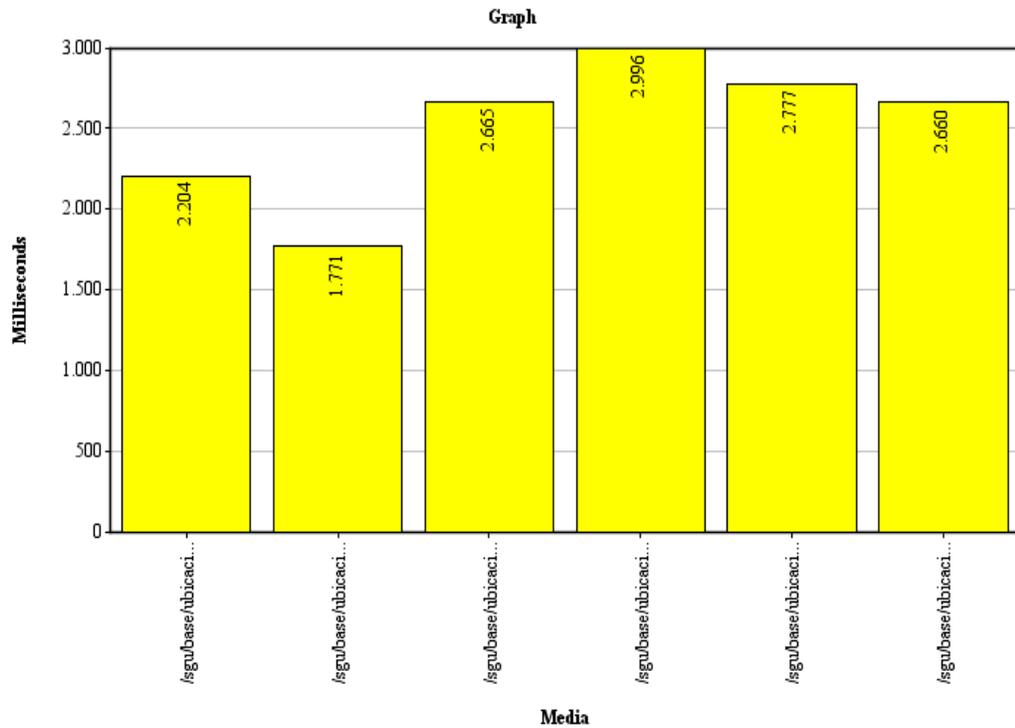


Figura 21: Resultado de las pruebas de rendimiento para 50 usuarios.